

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

DIJANA JARANOVIĆ

APLIKACIJA ZA PRAĆENJE KRVNOG TLAKA

Diplomski rad

Pula, 2018. godine

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko-komunikacijske tehnologije

DIJANA JARANOVIĆ

APLIKACIJA ZA PRAĆENJE KRVNOG TLAKA

Diplomski rad

JMBAG: 0303028531, redoviti student

Studijski smjer: Informatika

Predmet: Mobilne aplikacije

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Siniša Sovilj

Pula, ožujak, 2018. godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Dijana Jaranović, kandidatkinja za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, 28. ožujka 2018. godine



IZJAVA o korištenju autorskog djela

Ja, Dijana Jaranović dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Aplikacija za praćenje krvnog tlaka“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 28. ožujka 2018. godine

Potpis

Pula, 1. ožujka 2017.

DIPLOMSKI ZADATAK

Pristupnik: **Dijana Jaranović (0303028531)**

Studij: Sveučilišni diplomski studij Informatike

Naslov **Aplikacija za praćenje krvnog tlaka**

(hrv.):

Naslov Application for blood pressure monitoring

(eng.):

Opis Zadatak je razviti mobilnu Android aplikaciju namijenjenu bilježenju, pohrani
zadatka: i praćenju sistoličkog i dijastoličkog krvnog tlaka te srčane frekvencije kao osnovnih podataka u dijagnostici arterijske hipertenzije. Omogućiti i unos dodatnih podataka poput tjelesne mase, količine unesene soli, trajanja fizičke aktivnosti (informacije dohvatljive iz drugih aplikacija) i sl. Mobilnu aplikaciju razviti na hrvatskom jeziku te maksimalno pojednostavniti unos podataka i preglednost korisničkog sučelja.

Uz mobilnu razviti i poslužiteljsku Java web aplikaciju korištenjem Java Spring programskog okvira i Apache Tomcat poslužitelja. Nakon autentikacije web aplikacija treba omogućiti dohvat podataka s mobilnog uređaja putem REST API-a, njihovu pohranu u bazu poslužitelja te vizualizaciju podataka unutar internetskog preglednika.

Zadatak uručen pristupniku: 1. ožujka 2017.

Rok za predaju rada: 1. veljače 2018.

Mentor:

Siniša Sovilj

doc.dr.sc. Siniša Sovilj

SADRŽAJ

UVOD	1
1. OPIS I ANALIZA SUSTAVA	2
1.1. Hipertenzija kao zdravstveni problem današnjice	2
1.2. Ciljano tržište i analiza konkurencije	5
1.3. Motivacija za izradu aplikacije	9
1.4. Prednosti uvođenja aplikacije e-Tlak	10
1.5. Važnost korisničkog iskustva, korisničkog sučelja i dizajna aplikacija	11
2. API I BAZA PODATAKA APLIKACIJE E-TLAK	16
2.1. Razrada funkcionalnosti	16
2.1.1. <i>Klasni dijagram</i>	16
2.2. Implementacija i korištene tehnologije	17
2.2.1. <i>Programski okvir Node.js</i>	18
2.2.2. <i>Platforma Heroku i baza podataka mLab</i>	23
3. MOBILNA APLIKACIJA E-TLAK	25
3.1. Razrada funkcionalnosti	25
3.1.1. <i>Korisnički scenariji</i>	25
3.1.2. <i>Mobilna aplikacija Dnevnik krvnog tlaka</i>	28
3.2. Implementacija i korištene tehnologije	30
3.2.1. <i>Programski jezik Java</i>	31
3.2.2. <i>Operacijski sustav Android</i>	32
3.2.3. <i>IDE Android Studio i biblioteka Volley</i>	34
3.3. Korisničke upute	36
3.3.1. <i>Registracija i prijava</i>	37
3.3.2. <i>Unos krvnog tlaka i tjelesne mase</i>	37
3.3.3. <i>Prikaz dijagnoze</i>	38

3.3.4. Povijest unosa.....	39
4. WEB APLIKACIJA E-TLAK.....	40
4.1. Razrada funkcionalnosti	40
4.1.1. Korisnički scenariji.....	40
4.1.2. Prototip sučelja.....	42
4.2. Implementacija i korištene tehnologije.....	46
4.2.1. JavaScript, HTML i CSS jezici.....	46
4.2.2. Programski okvir Vue.js	49
4.3. Korisničke upute.....	54
4.3.1. Prijava.....	54
4.3.2. Pregled dijagnoze	55
4.3.3. Pregled krvnog tlaka	56
4.3.4. Pregled tjelesne mase.....	57
4.3.5. Pregled povijesti unosa	58
ZAKLJUČAK.....	60
LITERATURA.....	61
POPIS SLIKA.....	68
POPIS TABLICA	70
PRILOZI	71
SAŽETAK.....	72
SUMMARY	72

UVOD

U današnje vrijeme sve je više zdravstvenih problema, a samim time sve je veća potreba za sudjelovanjem tehnologije u samoj prevenciji tih problema. Arterijska hipertenzija predstavlja jedan od takvih problema koji obuhvaća posebnu skupinu korisnika koji imaju potrebu za što jednostavnijim načinom upotrebe tehnologije kao i posebnom preglednošću korisničkog sučelja. Aplikacija za praćenje krvnog tlaka naziv je diplomskog rada vezanog za predmet Mobilne aplikacije.

Rad se sastoji od nekoliko dijelova. Dio vezan za opis i analizu sustava obuhvaća istraživanje o hipertenziji kao zdravstvenom problemu današnjice u svijetu i Hrvatskoj, ciljano tržište aplikacije e-Tlak, analizu konkurencije i prednosti njenog uvođenja te ističe važnost korisničkog sučelja. Drugi dio je vezan za API i bazu podataka aplikacije e-Tlak, tj. razradu funkcionalnosti, implementaciju i korištene tehnologije. Dio vezan za mobilnu aplikaciju e-Tlak sadrži razradu funkcionalnosti, implementacije i korištenih tehnologija, korisničke upute za korisnika te usporedbu s mobilnom aplikacijom Dnevnik krvnog tlaka kao prvom verzijom. Dio vezan za web aplikaciju e-Tlak razrađen je na isti način. Opisi su detaljni kao i usporedba tehnologija, a priloženi su i razni dijagrami, prototipovi sučelja i snimke zaslona nastale kroz cjelokupni razvoj aplikacije. Svaki razvoj aplikacije ima svoje trajanje i proceduru te za svaki dio treba odabrati odgovarajuću tehnologiju. Prednje sučelje ili front-end obuhvaća kodiranje korisničkog sučelja, rukovanje događajima i interakcijama s poslužiteljem za dohvaćanje podataka. Back-end kao krajnji kraj se uglavnom bavi tehničkim stvarima koje nisu vidljive krajnjem korisniku, skriptiranjem na strani poslužitelja i interakcijom baze podataka kako bi aplikacije na front-endu mogle zatražiti potrebne podatke u mobilnoj ili web aplikaciji. Može se dogoditi da dođe do nekih nepredvidljivih stvari usred samog razvoja aplikacije, kao što je npr. otkrivanje puno bolje tehnologije za implementaciju pojedinog dijela aplikacije.

Cilj ovog diplomskog rada je sadržajno obuhvatiti navedenu tematiku na što bolji i jednostavniji način uz korištenje određene stručne literature i općeg znanja te na temelju iznesenih podataka, analizi i istraživanja doći do odgovarajućeg zaključka.

1. OPIS I ANALIZA SUSTAVA

Aplikacija e-Tlak namijenjena je za bilježenje, pohranu i praćenje sistoličkog i dijastoličkog krvnog tlaka te srčane frekvencije, odnosno osnovnih i glavnih podataka u dijagnostici arterijske hipertenzije. Arterijska hipertenzija predstavlja poremećaj u regulaciji krvnog tlaka od kojeg pati sve više ljudi u svijetu, ali i Hrvatskoj. Radi se o stanju trajno povišenog sistoličkog i/ili dijastoličkog krvnog tlaka; sistolički tlak koji predstavlja tlak koji se stvara prilikom otkucaja srca viši je od 140, a dijastolički tlak koji predstavlja tlak u krvnim žilama dok srce odmara je iznad 90. U pravilu, krvni tlak bi trebalo mjeriti kroz tri dana, barem po dva puta dnevno i to u različito doba dana (MSD priručnik, 2014). E-Tlak omogućuje unos i pregled izmjerenih vrijednosti krvnog tlaka kroz jednostavno korisničko sučelje i tako na jednostavan način omogućava korisnicima praćenje vlastitog zdravlja i vitalnih znakova. Aplikacija je na hrvatskom jeziku te je izrađena u obliku mobilne aplikacije i web aplikacije. Prvenstveno, njezin cilj je olakšati korisnicima-pacijentima, pa tako i samim liječnicima dijagnostiku, nadzor i prevenciju ove bolesti.

1.1. Hipertenzija kao zdravstveni problem današnjice

Procjene govore da oko 9,4 milijuna ljudi na svijetu umire zbog hipertenzije, točnije, to je oko 18% svih smrti na svjetskoj razini. Četiri od deset odraslih osoba u svijetu koje su starije od 25 godina ima povišeni krvni tlak. Polovica je onih koji znaju da imaju povišeni tlak, no ne poduzimaju ništa povodom toga, dok 50% osoba s hipertenzijom ne zna da ima povišeni krvni tlak (Kralj, 2016). Prema rezultatima istraživanja provedenih u Hrvatskoj, 58,6% ispitanika svjesno je svoje hipertenzije, od njih se liječi 48,4%, a samo 14,8% njih ima kontrolirani tlak. Predviđanja za 2025. godinu govore o porastu postotka hipertenzije na 29,2% svjetskog stanovništva (1,56 milijardi), a porast se odnosi i na stanovništvo muškog i ženskog spola (Erceg et al., 2013, p. 34). Brojke sve više rastu, a arterijska hipertenzija postaje eskalirajući svjetski zdravstveni problem.

Preventiva je vrlo važna kako bi se navedene brojke smanjile. Opće je poznato da je kod ove bolesti važno održavati optimalnu tjelesnu masu i zdravu i uravnoteženu prehranu, smanjiti unos soli i masti, ograničiti unos alkohola i prestati pušiti, biti tjelesno aktivan, no veliku važnost ima i samostalno bilježenje, prehrana i praćenje

krvnog tlaka, odnosno samokontrola. Lošom prevencijom i samokontrolom pacijenti povećavaju rizik od infarkta i moždanog udara, srčanih i bubrežnih bolesti i dr. Aplikacija e-Tlak pridonijela bi discipliniranom redovitom kontroliranju krvnog tlaka od strane pacijenta, ali i liječnika, s time i pravim terapijama s boljim učinkom. Vrlo je važno pratiti bolesnika i donositi potrebne odgovarajuće promjene. Svaki pacijent koji boluje od arterijske hipertenzije ima svoj tlakomjer i s tom bolešću treba imati mnogo strpljenja kao i liječnik. Najbitnije je voditi dnevnik izmjerenih vrijednosti krvnog tlaka, evidentirati tjelesnu masu, bilježiti izlaganja fizičkim opterećenjima u danima mjerenja i slične stavke. E-Tlak je suvremeni digitalni dnevnik koji pomaže pacijentu, a može pomoći i liječniku objektivnije analizirati ponašanje i oscilacije krvnog tlaka te samim time doprinosi boljoj, jednostavnijoj i kontinuiranoj zajedničkoj suradnji između pacijenta i liječnika, kroz mobilnu aplikaciju i web aplikaciju.

Od arterijske hipertenzije uglavnom pate bolesnici s dijabetesom, moždanim i vaskularnim bolestima, srčani bolesnici, pretile osobe, pušači, osobe pod stresom, najčešće između 50-65 godina, no i mlađi od 50 te mali broj adolescenata i djece (WHO, 2013, pp. 8-16). Uzimajući u obzir ove podatke, aplikacija e-Tlak obuhvaća muškarce i žene iz navedene skupine kao korisnike koji posjeduju pametni telefon i računalo, ali i liječnike. Pod korisnike mobilne i web aplikacije e-Tlak spadaju oboljeli koji žele naučiti samokontrolirati svoju hipertenziju te biti individualniji, sigurniji, i informiraniji. Također, podrazumijevaju se i oni koji su počeli poduzimati korake prevencije i žele pratiti napredak te oni koji, primjerice, imaju rizik oboljeti u budućnosti ili se bave fizičkom aktivnošću, pa povremeno prate stanje svog krvnog tlaka. Dio korisnika otpada i na starije osobe koje ne mogu toliko često posjećivati svog liječnika, pa žele kroz mobilnu aplikaciju evidentirati što više mjerenja svog krvnog tlaka kroz određeno razdoblje i onda posjetiti liječnika. Ukoliko žele detaljnije pogledati svoju dijagnozu, to mogu učiniti kroz web aplikaciju u čijem pregledavanju mogu sudjelovati i liječnici, medicinske sestre, treneri, ordinacije, medicinske škole, medicinski fakulteti i dr. Točnije, oni koji žele biti bolje upoznati s pacijentovim simptomima bolesti, točnim vrijednostima krvnog tlaka pojedinog bolesnika, upoznati bolje njegove životne navike, ostvariti aktivno sudjelovanje pacijenta i suradnju, kao i stvoriti kvalitetan individualan pristup prema svakome, saznati je li neka određena terapija stvarno neophodna i učinkovita i sl.

Posljednjih 20 godina, način dijagnostike arterijske hipertenzije i mjerenja krvnog tlaka se dosta promijenio. Do sada, krvni tlak se kontrolirao mjereći mehaničkim tlakomjerom dugotrajnim čekanjem ispred ordinacije liječnika te postupno automatskim i elektroničkim tlakomjerima u ljekarnama i kod kuće. To podrazumijeva i bilježenje dnevnika krvnog tlaka te terapiju lijekovima. Nažalost, dnevnik se uglavnom vodio na papiru, a znalo se i desiti da su pojedini pacijenti bili skloni uzimati terapiju samo kada bi im se krvni tlak naglo povišio, dok su neki od njih imali povišeni tlak samo u liječničkoj ordinaciji, dok u kućnim uvjetima nisu. Tu dolazi do povećanja krvnog tlaka kao posljedice tjeskobe od samog pregleda i straha pacijenta (ScienceDaily, 2017). U Hrvatskoj je sama edukacija pacijenata o ovoj bolesti slaba. Neki nisu svjesni vrijednosti krvnog tlaka, a dobar dio njih izbjegava piti lijekove jer tvrde da ih je mnogo. Neki od njih misle da im samo stres uzrokuje hipertenziju, pa da preko vikenda ili godišnjeg odmora ne moraju uzimati terapiju što je očiti znak da se pacijenti trebaju uključiti u savjetovanje i bolju samokontrolu ove bolesti (Meštrović, 2015). Mjerenje krvnog tlaka samo u ambulanti nije dovoljno jer je na taj način teško doći do odgovarajuće adekvatne terapije. Danas postoje suvremene metode, a svakako je jedna od njih da se krvni tlak kontinuirano mjeri kod kuće uz vođenje dnevnika, na dnevnoj, tjednoj ili mjesečnoj bazi, na temelju kojeg se može preciznije odrediti terapija. Koriste se pametni telefoni s mobilnim aplikacijama vezanim za mjerenje i praćenje krvnog tlaka od kojih većina nažalost nije precizna ni točna što može imati opasne posljedice. Dokazano je da su pacijenti samokontrolom snizili krvni tlak više nego uz uobičajenu skrb. U prvoj skupini pacijenata radi se o onima koji su provodili samokontrolu i samopraćenje, a u drugoj su se nalazili oni koji su primali uobičajenu skrb. Oni koji su provodili samokontrolu bili su osposobljeni za mjerenje tlaka i pridržavanje unaprijed određenog plana za donošenje odluka nakon različitih očitavanja tlaka (Kvantum-tim, 2014). Svakako, liječnik je ovdje vrlo bitan, a još bitnija je komunikacija između pacijenta i liječnika koja bi se puno više poboljšala suvremenim dnevnikom e-Tlak.

U nastavku slijedi tablica s kategorijama arterijskog tlaka po kojoj je rađena aplikacija e-Tlak. No, te granice su se oduvijek često pomicala i dan danas se govori o promijeni istih.

Tablica 1. Kategorije arterijskog tlaka

Kategorije	Sistolički arterijski tlak (mmHg)	Dijastolički arterijski tlak (mmHg)
Optimalan	< 120	<i>i</i> < 80
Normalan	120 – 129	<i>i/ili</i> 80 – 84
Visoko normalan	130 – 139	<i>i/ili</i> 85 – 89
Arterijska hipertenzija 1. stupnja	140 – 159	<i>i/ili</i> 90 – 99
Arterijska hipertenzija 2. stupnja	160 – 179	<i>i/ili</i> 100 – 109
Arterijska hipertenzija 3. stupnja	>= 180	<i>i/ili</i> >= 110
Izolirana sistolička hipertenzija	>= 140	<i>i/ili</i> < 90

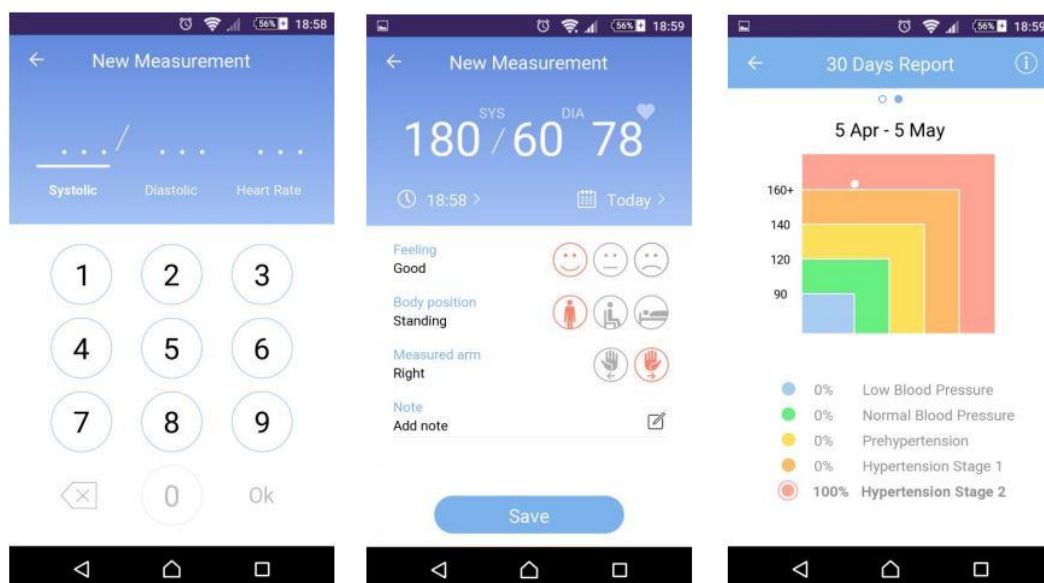
(Izvor: Horvat, 2016)

1.2. Ciljano tržište i analiza konkurencije

Ciljano tržište se odnosi prvenstveno na Hrvatsku; muškarce, žene, obitelji kao krajnje korisnike, pa i ambulante, domove zdravlja, poliklinike, bolnice, ljekarne i druge zdravstvene ustanove kao organizacije koje su uključene u komunikaciju, kao i na medicinsko obrazovanje u Hrvatskoj; fakultete, škole i udruženja. S vremenom, tržište bi se proširilo i na šire, Bosnu i Hercegovinu, Srbiju, pa možda i druge zemlje.

Ciljano tržište aplikacije treba uvijek detaljnije razložiti, odnosno opisati postojeća i konkurentna rješenja. Od boljih dnevnika krvnog tlaka u obliku mobilnih aplikacija ističu se IBP Blood Pressure, MedM Blood Pressure i Heartcare Blood Pressure. IBP Blood Pressure omogućuje praćenje i analizu krvnog tlaka uz interaktivne grafikone na dnevnoj, tjednoj, mjesečnoj i godišnjoj bazi gdje se uz pomoć klasičnih boja ukazuje na određenu dijagnozu hipertenzije. Iako ima dobru ocjenu, korisničko sučelje je obično, s klasičnim unosom i klasičnog dizajna što pojedine korisnike

možda ne privlači i ne zadržava koliko bi moglo. Također dobro ocijenjena aplikacija MedM Blood Pressure, ističe se jednostavnim unosom kroz pametnu tipkovnicu za razliku od prethodne. Prva aktivnost aplikacije odnosi se na pohranu gdje korisnik putem pametne „user-friendly“, odnosno korisniku prilagođene tipkovnice unosi svoj krvni tlak, tj. vrijednosti sistoličkog tlaka, dijastoličkog tlaka i pulsa. Nakon pohrane, slijedi opcionalni daljnji unos korisnika gdje korisnik može unijeti kako se osjeća, unijeti poziciju tijela, ruku na kojoj je krvni tlak izmjeren i neke dodatne bilješke. Inače, početna stranica prikazuje posljednje unose, uglavnom tri posljednja mjerenja krvnog tlaka. Navedena aplikacija ima i zaslon za praćenje i izvještavanje o krvnom tlaku gdje je moguć pregled unosa i statistika u rasponu primjerice, 30 dana, što je moguće u obliku dvije vrste grafikona. Pregled grafikona je moguć na dnevnoj, tjednoj i mjesečnoj bazi. Postoji i zaslon vezan za povijest koji obuhvaća sve unose krvnog tlaka, po danima i vremenu unosa. Što se tiče aplikacije Heartcare Blood Pressure, ona ima više mogućnosti, ali je zbog toga izgledom i malo kompleksnija. Izbornik je duži i s više opcija te je način unosa vrijednosti krvnog tlaka klasičan. Korisničko sučelje sastoji se jednim dijelom od kartica koje isto tako predstavljaju izbornik. Nakon unosa vrijednosti, otvara se prozorčić koji javlja dijagnozu i predlaže unos daljnjih simptoma, osjećanja i sl. Iako ima bolju ocjenu od prethodne aplikacije, korisničko sučelje je lošije i otežan je unos. Spomenute mobilne aplikacije su plavog, „medicinskog“ dizajna (Google Play, 2018). Slika u nastavku prikazuje nekoliko zaslona mobilne aplikacije MedM Blood Pressure.



Slika 1. Glavni zasloni mobilne aplikacije MedM Blood Pressure
(Izvor: Google Play, 2018)

Što se tiče web aplikacija i novijih konkurenata u Hrvatskoj, tu je Medikor. Radi se o aplikaciji koja omogućuje kontrolu i evidenciju krvnog tlaka i srčanog pulsa, glukoze u krvi, težine, indeksa tjelesne mase te udjela masnog i mišićnog tkiva u organizmu. Dostupna je na web stranici putem svih vrsta računala i pametnih telefona. Nakon registracije i prijave, korisnik dobiva smjernice o pravilnom načinu mjerenja, kao i unosu i čitanju rezultata. Aplikacija se uglavnom sastoji od klasičnih formi za unos. Rezultati se spremaju u dnevnik mjerenja koji sadrži grafički prikaz koji se može ispisati ili poslati na mail, primjerice, liječniku. Preporučuje se uz poznate tlakomjere Omron i Beurer te postoji stotinjak ljekarni diljem Hrvatske s Medikor stanicama za provjeru zdravlja. Medikor je zapravo sastavni dio stanice; bit je nakon mjerenja u ljekarni unijeti vrijednosti u Medikor aplikaciju. U nastavku je prikazan izgled dnevnika mjerenja u web aplikaciji Medikor (Medikor, 2016).



Slika 2. Glavni zaslon web aplikacije Medikor
(Izvor: Medikor, 2016)

Osim spomenutog konkurenta, Omron kao najpoznatiji i najugledniji proizvođač tlakomjera na svijetu, uz tlakomjer nudi mogućnost povezivanja s pametnim telefonom putem mobilne aplikacije Omron Connect. Aplikacija omogućuje brzi prikaz izmjerenih vrijednosti krvnog tlaka, pregled grafikona i povijesti, prosjek izmjerenih vrijednosti kroz određeni vremenski period, lakše pregledavanje, spremanje i dijeljenje zdravstvenih podataka s aplikacijama poput Apple Health ili uz osobno predočenje liječniku. No, ima slabiju ocjenu i malo korisnika (Google Play, 2018). Svaki konkurent se ističe svojim značajkama i prednostima, ali ima i svoje nedostatke. Dnevnik krvnog tlaka kao rješenje treba imati što više olakšan unos vrijednosti i jednostavno korisničko sučelje kao glavnu posebnost kojom će se isticati, što e-Tlak ima.

1.3. Motivacija za izradu aplikacije

Kako bi se još preciznije odredila budućnost aplikacije e-Tlak, dobro je napraviti SWOT¹ analizu koja obuhvaća analizu okoline, raznih karakteristika okruženja i važnih strateških čimbenika, odnosno snaga, slabosti, prijetnji i prilika okruženja u kojem će e-Tlak djelovati.

Jaka točka e-Tlaka je to što se radi o jedinoj hrvatskoj verziji aplikacije za bilježenje, pohranu i praćenje krvnog tlaka. U odnosu na konkurenciju koja velikim dijelom obuhvaća strane mobilne aplikacije vezane za mjerenje i evidenciju krvnog tlaka, aplikacija e-Tlak jedina podrazumijeva i mobilni i web aspekt uz prilagođeno jednostavno korisničko sučelje te je jedina vezana za hrvatske pacijente i liječnike gdje nema takve konkurencije. Hrvatski liječnici će zasigurno podržati e-Tlak kao aplikaciju jer je hrvatski proizvod i usluga, a pacijenti će ju prihvatiti jer je na hrvatskom jeziku te jer mobilna i web aplikacija bolje povezuju pacijenta i liječnika u ordinaciji što predstavlja značajku koju konkurencija nema. Snaga koja se dobro može iskoristiti je i to što je aplikacija poput e-Tlaka veoma potrebna među građanima s obzirom na to da se želi probuditi svijest o arterijskoj hipertenziji. Što se tiče slabosti, postoji mogućnost da ne posjeduju svi pacijenti pristup internetu, računalo i odgovarajući pametni telefon koji bi podržavao mobilnu aplikaciju e-Tlak, pa na starijim modelima ona neće biti podržana. Neki stariji pacijenti niti nemaju pametni telefon, jedan dio njih neće htjeti prihvatiti bilježiti svoj krvni tlak putem mobilnog uređaja, stoga tu može doći do izbjegavanja. Što se tiče liječnika, tu nema problema jer svaka ordinacija ima pristup internetu i računalo kao glavne predispozicije potrebne za pregledavanje mobilne i web aplikacije vezano za objektivnije i dodatno razmatranje dijagnoze pacijentovog krvnog tlaka. Još jedna slabost je ta što je konkurencija kao što je Omron osvojila dio tržišnog segmenta, na način da uz tlakomjer nudi i mobilnu aplikaciju. Iako, to može biti i snaga jer će se korisnici drugih tlakomjera koji ne uključuju mobilnu aplikaciju, odlučiti na korištenje e-Tlaka. Može se dogoditi i da pacijent želi koristiti mobilnu aplikaciju e-Tlak i pokazati detaljnije svoju dijagnozu na web aplikaciji, a njegov liječnik ne želi ili nema vremena detaljnije pregledati dijagnozu kroz web aplikaciju, i obrnuto. Velika je prilika e-Tlaka da sudjeluje u zdravstvenom sustavu Hrvatske, da poveže pacijenta i liječnika te da osvijesti ljude o arterijskoj hipertenziji i time smanji ostale bolesti

¹ engl. Strengths, Weaknesses, Opportunities and Threats

povezane s hipertenzijom. S obzirom na to da se Hrvatska nastoji što više modernizirati sustavima kao što su e-Građani, e-Dnevnik i drugi, ovo je odlična šansa za nadogradnju toga. Osim u kućanstvima i hrvatskim ordinacijama, postoji prilika širenja i u bolnicama, fakultetima i izvan granica Hrvatske. Ono što može biti prijetnja u smanjivanju udjela e-Tlaka na tržištu jesu svjetski konkurenti ukoliko postanu previše poznati i korišteni u svijetu, privatnim klinikama i sl. Što se tiče potražnje od strane drugih država u svijetu, tu se hrvatska verzija neće prihvatiti što zahtijeva promjenu jezika aplikacije e-Tlak, iako se ona tome može vrlo lako prilagoditi. Takva promjena ne bi štetila ovom proizvodu jer bi postojala verzija na hrvatskom i npr. verzija na engleskom, odnosno opcija odabira jezika u aplikaciji.

1.4. Prednosti uvođenja aplikacije e-Tlak

Mnogo je prednosti i koristi koje će se ostvariti uvođenjem rješenja kao što je aplikacija e-Tlak. Korisnici kao pacijenti će kroz mobilnu i web aplikaciju naučiti samostalno pratiti svoj krvni tlak kod kuće čime će stvoriti realnu sliku o svom zdravlju te je sami naučiti spoznati. To je izvrsna dopuna i sigurnost prilikom odlaska liječniku na potvrdu dijagnoze, pogotovo za one koji imaju nekontrolirani i komplicirani krvni tlak. E-tlak će uz informacije vezane za mjerenje omogućiti korisniku bolju i odgovorniju prevenciju ove bolesti. Većina ljudi koji boluju od arterijske hipertenzije ima svoj tlakomjer, pa će im e-Tlak biti dobra nadopuna kao dnevnik. Čak i oni koji nemaju tlakomjer, ukoliko mjerenje obave kod svog liječnika ili u ljekarni, moći će izmjerene vrijednosti evidentirati kroz e-Tlak na svom pametnom telefonu. Više neće biti gubitaka papira na kojem se klasično vodio dnevnik krvnog tlaka, kao ni samim time krivog iščitavanja jer će ova aplikacija omogućiti digitalno sigurno i jednostavno pohranjivanje te dohvaćanje podataka o krvnom tlaku. Za starije pacijente će članovi obitelji moći unositi vrijednosti u mobilnu aplikaciju i na taj način priložiti relevantne podatke prilikom zajedničkog posjeta liječniku. Uz pacijente s astmom i dijabetesom koji odgovorno kontroliraju svoju bolest, to bi postali i pacijenti s hipertenzijom. Preuzimanje odgovornosti pacijenta za vlastito stanje vrlo dobro djeluje na njega i osnažuje ga. Tako će e-Tlak potaknuti pacijenta da za sebe čini najviše što je u njegovoj moći.

S obzirom na to da će web aplikacija e-Tlak korisniku nuditi detaljniju vizualizaciju dijagnoze krvnog tlaka, to će ujedno omogućiti i liječniku bolji pregled pacijenta, uzvratanje prave povratne informacije i sigurniju promjenu terapije lijekovima. Uz ovakvu suvremenu digitaliziranu prevenciju, zasigurno bi pao broj bespotrebnih hospitalizacija pacijenata koji (ne)boluju od hipertenzije, spriječio bi se nastanak moždanog i srčanog udara kao i ostalih bolesti, pa i mortaliteta povezanog s hipertenzijom, dok bi se povećalo promicanje zdravlja. Osim što će e-Tlak pridonijeti boljoj suradnji između pacijenta i liječnika kao krajnjih korisnika, pridonijet će i boljem funkcioniranju organizacija, tj. sustava zdravstvene skrbi jer upravo ovakvo pravilno osposobljavanje i praćenje te nadzirana kućna kontrola tome doprinosi kao i manje opterećenim zdravstvenim službama. Hrvatski građani često idu previše kod liječnika te čekaju pred ordinacijom samo kako bi dobili recept za svoju hipertenziju, često iz neznanja završe kod liječnika, a e-Tlak će to smanjiti i osloboditi liječniku vrijeme za ozbiljnije slučajeve. Neki ne vide svog liječnika po nekoliko mjeseci, pa je ovo dobar način evidencije krvnog tlaka npr. na mjesečnoj bazi. Sažeto, omogućit će se bolje definiranje očekivanja i plana liječenja, praćenje pacijenta i promjena, doći će do boljeg zadovoljstva pacijenata i individualizacije te liječnikovog boljeg posvećivanja pacijentima kojima je to potrebnije i pacijentima s posebnom dijagnostikom hipertenzije.

1.5. Važnost korisničkog iskustva, korisničkog sučelja i dizajna aplikacija

Prije samog shvaćanja aplikacije e-Tlak dobro je razumjeti razliku između pojmova korisničko iskustvo i korisničko sučelje. Korisničko iskustvo² predstavlja iskustvo koje proizvod, u ovom slučaju aplikacija, stvara ljudima kao korisnicima za vrijeme uporabe što uključuje njihovo ponašanje, stav i doživljaj prema aplikaciji. To je korisnikovo očekivanje i osjećaj koji ima dok koristi aplikaciju. Ono određuje na kraju krajeva radi li se o uspješnoj aplikaciji ili promašaju. Tu se ne radi o tome kako aplikacija funkcionira iznutra, već izvana u svakodnevicu: navigiranje kroz aplikaciju, pritiskanje gumba, unos podataka i sl. Glavnina navedenog se događa na korisničkom sučelju, stoga je ono vrlo važno kao i način na koji je dizajnirano.

² engl. User Experience, UX

Korisničko sučelje³ predstavlja poveznicu koja obuhvaća cjelokupni doživljaj tj. interakciju između aplikacije i korisnika. Dizajn korisničkog sučelja je danas iznimno važan iz razloga što ono treba rješavati osnovne probleme i jasno prenijeti poruku korisniku aplikacije. Dobro dizajnirano korisničko sučelje je ono koje omogućuje da aplikacija radi ono za što je namijenjena i tako stvara dobro korisničko iskustvo (Garrett, 2011, pp. 3-11).

Glavni i osnovni fokus treba biti na korisnika, njegove zadaće te treba omogućiti korisnu upotrebu aplikacije u što kraćem vremenu. Upotrebljivost svake aplikacije ovisi o kvaliteti implementacije (izgled i funkcionalnost korisničkog sučelja), a vrlo vjerojatno npr. i o tipu ciljanog mobilnog uređaja; različiti „osjećaji ugođe“ kod korištenja uređaja s fizičkom tipkovnicom i uređaja sa zaslonom osjetljivim na dodir (Ljubić, 2011, pp. 3-6). Sučelje treba biti dizajnirano s korisnikovog gledišta, ne treba komplicirati i treba izbjegavati dodatne probleme za korisnika. Kroz korisničko sučelje treba ostvariti konzistentnost, tj. mogućnost da će ga korisnik lako naučiti, upamtiti i u sljedećoj upotrebi znati ponovno koristiti. Korisničko sučelje treba biti jednostavno, smisleno i funkcionalno istovremeno. Treba biti „prijateljski naklonjeno“ korisniku, samoobjašnjivo i očito. Dizajn prečaca prvi privlači pažnju korisnika i stvara prvi dojam. Ukoliko se korisniku na prvi pogled ne sviđa prečac, vjerojatno će odmah i izgubiti interes za preuzimanjem aplikacije. Dodavanjem nepotrebnih detalja u službi vizualne atrakcije, korisničko sučelje će postati manje jasno, dok će dodavanjem smislenih detalja u službi sadržaja sučelje postati jasnije. Svaki element korisničkog sučelja koji je nepotreban, valja ukloniti. Sučelje bi trebalo pratiti konvencije platforme. Ako ono funkcionira na drugi način od uobičajenog sučelja platforme na koju je korisnik navikao, postoji mogućnost da će korisnik odustati od aplikacije zbog nesnalažljivosti. Također, treba imati na umu da sučelje mobilne aplikacije nije isto što i sučelje web aplikacije. Web aplikacije mogu sadržavati velike logotipove i razne elemente, dok kod mobilnog sučelja to nije pametno jer oduzima prostor za informacije i može predstavljati prepreku korisniku (Janjanin, 2011). Što se tiče tipki kao dijelu korisničkog sučelja na zaslonu osjetljivom na dodir, ciljane površine trebale bi biti veličine vrha prsta, a elementima sučelja bi trebao biti osiguran odgovarajući razmak kako bi se izbjegle greške i vizualno prikrivanje aplikacijskog sadržaja prilikom odabira. Inicijalna istraživanja pokazala su da je, zbog antropoloških

³ engl. User Interface, UI

karakteristika samog čovjeka, poželjna veličina interaktivnog objekta na zaslonima osjetljivima na dodir 22 mm (Lee et al., 2009, pp. 309-318). Općenito, treba uskladiti boje sučelja, kao i tekst, odnosno veličinu i poravnanje teksta. Korištenje animacija i njihovo izvršavanje u početku je korisniku zanimljivo, no s vremenom svako pokretanje animacije oduzima vrijeme korisniku i usporava samu aplikaciju. Vrlo važna je brzina pomicanja i brzina animacije. One imaju svrhu samo onda kada služe za komunikaciju, no ako su same sebi svrha nemaju smisla. Vrlo je važno uskladiti količinu informacija na zaslonu: izbaciti nepotrebne informacije i one važne ravnomjerno rasporediti na zaslonima aplikacije.

Treba dobro poznavati korisnike aplikacije i njihove karakteristike. U slučaju e-Tlak aplikacije, veći dio korisnika spada pod zrelu i stariju dobnu skupinu. Kod njih postoje problemi s vidom, motorikom i sluhom. Što se tiče vida osim problema s dioptrijom, starije oči primaju samo 1/3 svjetla u odnosu na mlađe; stvara se žućkasta boja u vidnom polju što im otežava razlikovanje nijansi plave ili ljubičaste boje. U tom slučaju, kod korisničkog sučelja treba izbjegavati veličinu fonta manju od 16 piksela te treba koristiti lako čitljivu vrstu fonta. Bitno je povećati elemente sučelja kao i ključne gumbe. Bilo bi dobro izbjegavati plavu boju za važne elemente sučelja i smanjiti udaljenost između elemenata sučelja koji će se koristiti u slijedu poput polja obrazaca, ali paziti da je razmak između njih najmanje dva milimetra (Campbell, 2015). Svakako, ovi korisnici žele razumjeti modernu tehnologiju i izume, žele ići ukorak s drugima te je krivo uvjerenje da izbjegavaju tehnologiju. Spremni su i mogu naučiti komplicirano ukoliko uvide pravu korist od toga. Što se tiče motorike, oni koji imaju preko 65 godina trebaju oko 50 do 100% više vremena kako bi dovršili zadatak, u odnosu na mlađe od 30 godina. Rezultati istraživanja američke kompanije Medium u suradnji s Češkim tehničkim sveučilištem u Pragu pokazuju to da bi 48% ljudi iznad 50 godina cijenili veći font na mobilnom uređaju, a 80% njih misli da njihov uređaj ima previše mogućnosti nego što će oni ikad iskoristiti. Dokazano je i da se spomenuti korisnici bolje i brže služe zaslonima osjetljivim na dodir nego računalnim mišem. Iako se za njih ne preporučuje mobilni uređaj, to se uveliko može olakšati primjerenim dizajnom mobilnog korisničkog sučelja. Dakle, korisničko sučelje za navedene korisnike treba biti što jednostavnije i što lakše za korištenje. Ukoliko je korisničko sučelje komplicirano, velika je mogućnost da će korisnik kriviti sebe, a ne tehnologiju. Kako bi se prilagodilo motorici, osim prije spomenutog pravila za razmak

između elemenata sučelja, trebalo bi postaviti gumbе za navigaciju na sve zaslone, da su prisutni i vidljivi. Takav korisnik želi znati što će se dogoditi nakon što pritisne određeni gumb, pa je katkad uz gumb dobro imati kratke tekstualne oznake. Što se tiče korisnika s oslabljenim sluhom, dobro je u aplikaciju ubaciti zvuk obavijesti dužeg trajanja. Primjerice, nakon što je krvni tlak pohranjen, dati povratnu informaciju zvukom uz veću obavijest centriranu na zaslonu, što je slučaj kod mobilne aplikacije e-Tlak. Svaki komplicirani zaslon treba imati kratku i jasnu instrukciju te treba izbjegavati „slideanje“ i „skrollanje“ (Slaviček, 2014). Bitnu ulogu ima i dobar kontrast, topliji tonovi, lako upamtljivi i jednostavni gumbi u obliku ikona umjesto klasičnih tekstualnih, efekti pritisnutog i selektiranog, tekst uz gumbе ukoliko je moguće i potrebno i sl. Na kraju krajeva, svaki korisnik će griješiti prilikom korištenja aplikacije te treba olakšati korisniku ispravak što brže i lakše, npr. krivog unosa (Loureiro et al., 2014, pp. 42-46). Stariji korisnici vole da im je aplikacija od koristi, da im pojednostavi svakodnevicu i da ima stvarnu svrhu u njihovom životu. Godine su samo broj te starost nije nesposobnost.

Što se tiče web aplikacija, korisniku treba objasniti i ukazati na glavne interakcije koje aplikacija nudi, postaviti očekivanja i istaknuti glavne poveznice te uključiti skočne prozore kao brzu povratnu i zamjetljivu informaciju. Kao i kod mobilnih aplikacija, kod web aplikacija je još važnije izbjeći pogreške te ako se one dogode imati najlakši način ispravka. Isto tako treba pažljivo razmisliti o mjestu elemenata na sučelju web aplikacije, njihovoj veličini te se treba držati standarda. Cilj je pobrinuti se da korisnik lako nauči koristiti se web korisničkim sučeljem jer je danas kod web aplikacija glavni fokus na informaciju i njeno predočenje. Suvremeno web korisničko sučelje je puno vizualnih interakcija, dizajn interakcija i navigacije je napredan i dinamičan uz pravilnu upotrebu boja te SEO⁴ optimizacija sve više ima važnu ulogu (Moore Williams, 2017). Što se tiče kontrolnih ploča, izvještaja i grafikona, treba ih pravilno vizualizirati, na jednostavan način istaknuti upečatljivim bojama i omogućiti korisniku da bira pregled koji želi (Pal, 2016).

Sve su više aktualna korisnička sučelja koja su prilagođena korisnicima. Takva sučelja su prilagođena korisnicima te obuhvaćaju jednostavan unos kroz pametnu tipkovnicu, preglede i prikaze kroz dinamičke tablice i sl. To omogućuje još lakše korištenje cijele aplikacije, upravljanje podacima i evidencije. Korisničko sučelje koje

⁴ engl. Search Engine Optimization – optimizacija web stranica za tražilice

je u tom pogledu komplicirano, puno teksta i teško za pamćenje i korištenje, brzo će dovesti do toga da aplikacija bude zaobiđena, dok korisničko sučelje prilagođeno korisnicima omogućuje da se izgradi korisnikovo pozitivno iskustvo. Ono što je nekada bilo teško, danas je kroz navedenu vrstu korisničkog sučelja olakšano korisniku i on to čini sa zadovoljstvom. Kod e-Tlaka korisničko sučelje je „elder-friendly“, tj. prilagođeno starijim korisnicima. Primjerice, kod unosa vrijednosti krvnog tlaka, pametna tipkovnica automatski prelazi u sljedeće polje bez da korisnik mora odabirati svako polje posebno. Na isti način funkcionira i brisanje vrijednosti.

Nakon dobrog istraživanja korisnika, bilo bi poželjno „pokusno“ isprobati aplikaciju na korisniku, pa kroz to saznati što je potrebno nadodati ili ispraviti kod korisničkog sučelja. Upotrebljivost sučelja aplikacije dovodi korisnika do mogućnosti učenja, subjektivnog zadovoljstva i lakoće korištenja. Može se reći da je dobar dizajn kada je sučelje aplikacije takvo da mu dizajner nema što više oduzeti, a ne kada nema što dodati (Jaranović, 2015, pp. 12-14). Treba uzeti i u obzir da će se stariji korisnici sve više modernizirati. Općenito, svaki čovjek stari; neki će se znati koristiti tehnologijom, neki manje te aplikacije treba prilagoditi godinama i mogućnostima kao i dizajnirati korisnička sučelja za ljude i budućnost. Sve to čeka u budućnosti i uz prilagođeno korisničko sučelje sve postaje brže i lakše. Što će se dogoditi s onima koji danas drže pametne telefone u rukama, kada ostarje? O njihovoj sudbini jednostavnog korištenja korisničkog sučelja odlučuju i omogućuju dizajneri. Nekad onaj dizajn koji se radi po intuiciji, slobodom, čini korisničko sučelje poznatim, posebnim i privlačnim za korištenje.

2. API I BAZA PODATAKA APLIKACIJE E-TLAK

Kako bi se razumjelo funkcioniranje cijelog sustava e-Tlak, najbolje je prvo shvatiti funkcioniranje API⁵-ja i baze podataka. Sustav e-Tlaka čini mobilna aplikacija e-Tlak, web aplikacija e-Tlak, API i baza podataka. Kako bi se mobilna i web aplikacija povezale i ostvarile komunikaciju, potrebno je izraditi API i bazu podataka. API obuhvaća funkcije, procedure i rutine koje omogućuju interakciju s nekom softverskom komponentom ili sustavom, odnosno njegovim značajkama i podacima. Baza podataka predstavlja djeljivu kolekciju međusobno logički povezanih podataka. U ovom slučaju, API omogućuje mobilnoj aplikaciji e-Tlak i web aplikaciji e-Tlak komunikaciju s bazom podataka, dok baza podataka zaprima zahtjeve zatražene kroz mobilnu i web aplikaciju i isporučuje rezultate putem API-ja. Ovo poglavlje se detaljnije odnosi na poslužiteljsku stranu aplikacije, tj. back-end dio što podrazumijeva određene tehnologije i alate te specifičan postupak implementacije.

2.1. Razrada funkcionalnosti

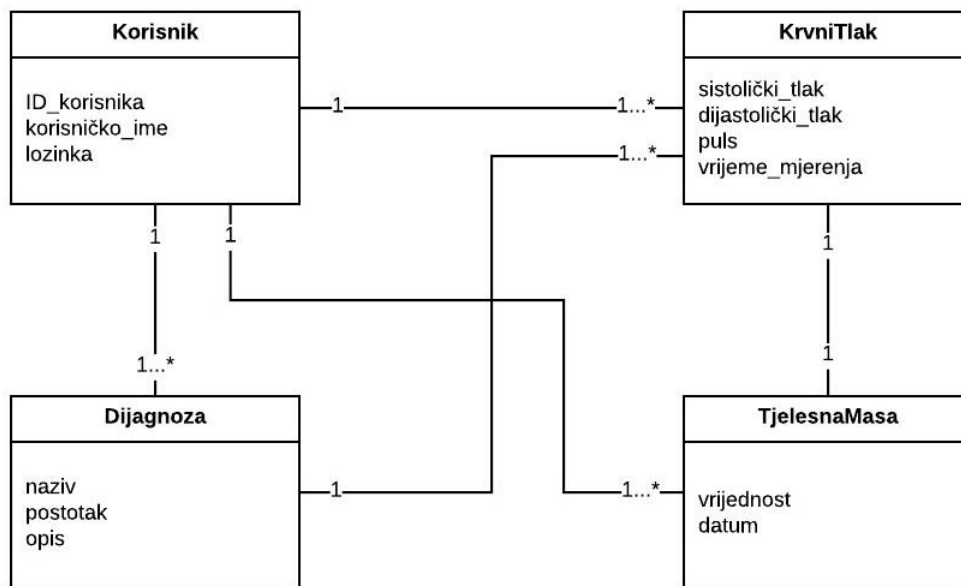
2.1.1. Klasni dijagram

Kako bi se bolje opisala struktura cijelog sustava e-Tlak, sastavljen je klasni dijagram⁶. On sažeto prikazuje glavne klase te njihove glavne atribute i međusobne veze.

Glavne klase sustava e-Tlak su: Korisnik, KrvniTlak, TjelesnaMasa i Dijagnoza. Atributi klase Korisnik su: ID_korisnika, korisničko_ime i lozinka. Klasa KrvniTlak ima sistolički_tlak, dijastolički_tlak, puls i vrijeme_mjerenja kao atribute. Što se tiče klase TjelesnaMasa, njezini atributi su vrijednost i datum, dok klasa Dijagnoza ima atribute: naziv, postotak i opis. Navedene klase i atributi, uz međusobne veze, prikazani su na slici u nastavku.

⁵ engl. Application Programming Interface – aplikacijsko programsko sučelje

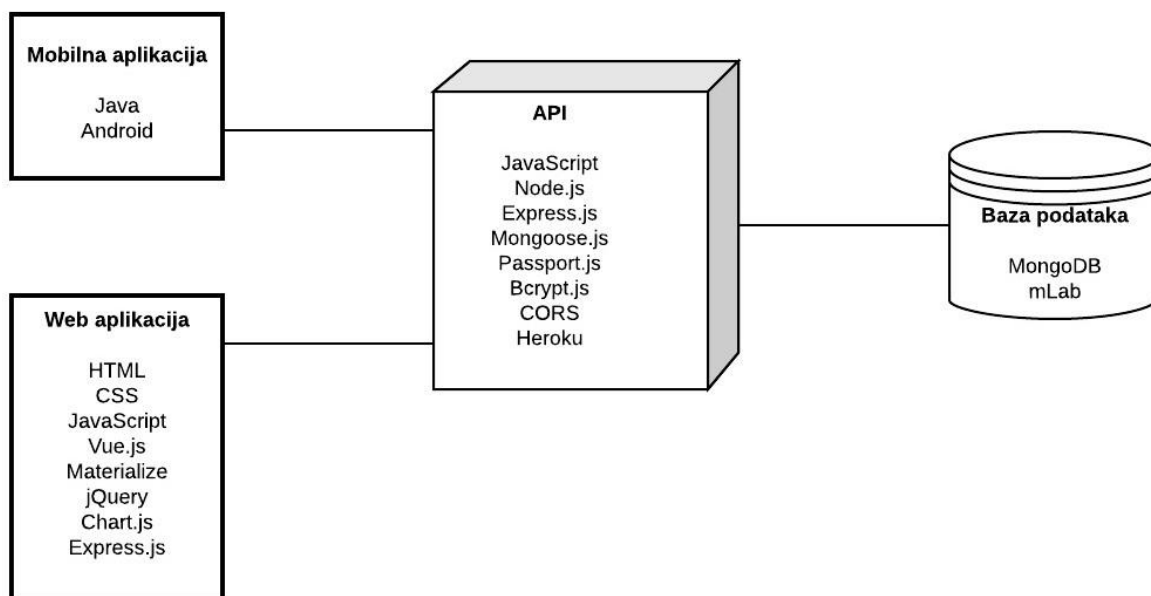
⁶ engl. Class diagram



Slika 3. Klasni dijagram sustava e-Tlak
(Izvor: autorica, 2018)

2.2. Implementacija i korištene tehnologije

Svaki dio sustava e-Tlak ima svoju specifičnu arhitekturu. Za implementaciju API-ja koristio se programski jezik JavaScript te glavne tehnologije Node.js, Express.js, Mongoose.js, Passport.js, Bcrypt.js, CORS i Heroku. Bazu podataka sustava čine MongoDB i mLab. Mobilna aplikacija e-Tlak implementirana je uz pomoć programskog jezika Java i operacijskog sustava Android. Za implementaciju web aplikacije e-Tlak korišteni su JavaScript, HTML, CSS, Vue.js, Materialize, jQuery, Chart.js i Express.js. U nastavku će se pojasniti svaki od jezika, tehnologija i alata vezanih za API i bazu podataka, a kasnije i oni vezani za mobilnu aplikaciju i web aplikaciju. Na slici je prikazan dijagram komponenata cijelog sustava e-Tlak.



Slika 4. Dijagram komponenata sustava e-Tlak
(Izvor: autorica, 2018)

Jedna od solucija za API mogao je biti REST⁷ API. REST je skup pravila za razvoj aplikacija koji omogućuju specifičnu implementaciju i održavanje. REST se obično koristi u kombinaciji s HTTP⁸ protokolom koji nudi sučelje za implementaciju RESTful API-ja. Glavne prednosti koje HTTP omogućuje u REST arhitekturi su uniformno sučelje i privremena pohrana podataka u memoriju (Brdar, 2016, pp. 3-14). No, pronašao se suvremeniji odabir tehnologije koji se pokazao kao bolje i puno jednostavnije rješenje u slučaju aplikacije e-Tlak koje ne zahtijeva previše rada oko implementacije, što je detaljnije argumentirano u nastavku rada kao i samom zaključku.

2.2.1. Programski okvir Node.js

Node.js je platforma za izradu poslužiteljskih aplikacija u JavaScript programskom jeziku. To znači da cijela aplikacija može biti realizirana uz upotrebu jednog programskog jezika, odnosno omogućeno je korištenje jedinstvenog jezika između front-enda i back-enda. Node.js osmislio je Ryan Dahl. On je početkom 2009. godine pokušao implementirati sustav koji će signalizirati svojim korisnicima događaj

⁷ engl. Representational State Transfer

⁸ engl. HyperText Transfer Protocol

na poslužiteljskoj strani. Nakon istraživanja zaključio je da jedino JavaScript programski jezik nema predefinirani set ulazno/izlaznih rutina te mu omogućava da implementira vlastite ideje i ubrza sam sustav (Penezić, 2013). Node.js je skalabilan i brz te nudi jednostavan i asinkroni event-driven (upravljan događajima) model programiranja. Izdan je pod MIT licencom, odnosno licencom otvorenog koda koja dozvoljava ponovno korištenje softvera. Također, može se koristiti na svim platformama: Windows, Linux i Mac. Primjeri kod kojih se Node.js koristi su: chat (razgovori), za izradu poslužiteljskog Proxy servisa (posrednik između korisnika i glavnog poslužitelja), za tok podataka kod tradicionalnih web platformi, kod unosa gdje se Node.js prilagođava svim količinama nadolazećeg podatkovnog prometa i sl. Mnoge poznate informatičke kompanije temelje svoje aplikacije na radu u pomoćnom alatu Node.js, kao što su: IBM, Microsoft, LinkedIn, Netflix, SAP, Cisco Systems, PayPal i dr (Raguž, 2017, pp. 1-11).

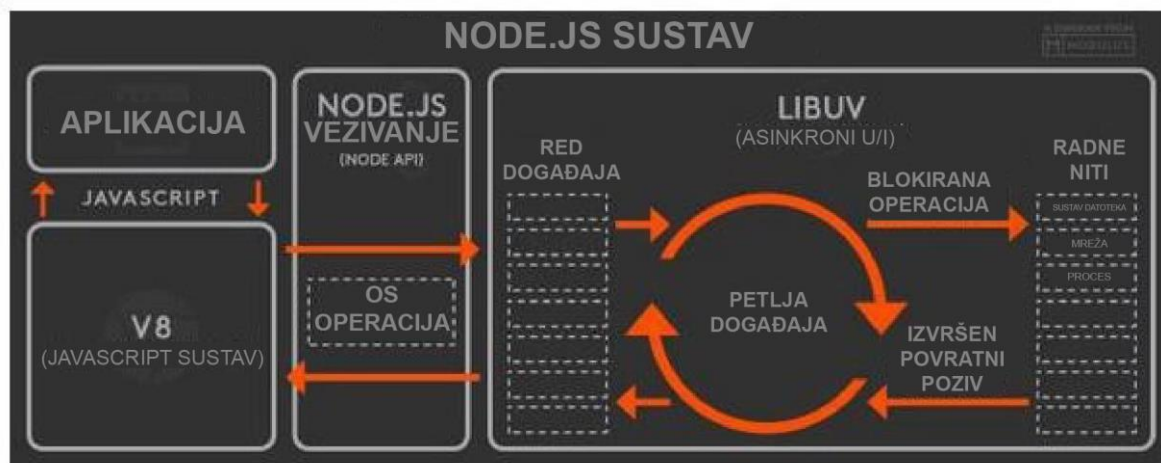
Node.js arhitekturu čine dvije glavne komponente:

- jezgra i
- moduli.

Jezgra je izrađena u programskim jezicima C i C++. Node.js je nastao na temeljima Google V8 JavaScript sustava uz dodatak vlastitog koda te implementaciju web poslužiteljskog dijela. To podrazumijeva i Libuv biblioteku i veze protokola. Google V8 JavaScript je zapravo JIT⁹ prevoditelj JavaScripta. Biblioteka Libuv obrađuje asinkrone ulazno/izlazne operacije i glavnu petlju (Chandrayan, 2017). Općenito, programiranje se obavlja sinkrono; linija koda se izvršava, sustav čeka rezultat, rezultat se procesuiri i zatim se izvršavanje programa nastavlja. Kod JavaScripta u slučaju izvršavanja nekih sporijih ulazno/izlaznih operacija, kao što je čitanje podataka iz baze podataka, program ne čeka, nego ide na izvršavanje sljedeće linije koda. Kada se ulazno/izlazna operacija vrati, pokreće se funkcija povratnog poziva i rezultat se procesuiri (Tošić, 2015). Zahvaljujući Google Chrome V8 JavaScript sustavu, radi se o iznimno brzom izvršavanju programskog koda. Node.js se arhitekturno savršeno uklapa sa zahtjevima izrade modernih „real-time“ web aplikacija ili web aplikacija u realnom vremenu. Dakle, radi se o korištenju novih i bržih načina rada sa ulazno/izlaznim sustavima te rad u jednoj petlji sa signalizacijom događaja. Cijela platforma temeljena je na događajima, a ne na nitima. S obzirom na

⁹ engl. Just-In-Time - „točno-na-vrijeme“

to da u sebi sadrži web poslužiteljsku podršku, za rad mu nije potrebno nikakvo drugo web poslužiteljsko programsko rješenje. Na slici u nastavku prikazana je pojašnjena arhitektura.



Slika 5. Arhitektura platforme Node.js
(Izvor: Chandrayan, 2017)

Kako bi se smanjila veličina koda koji se interpretira, uz Node.js instalaciju dolazi i ugrađena podrška za upravljanje paketima pomoću NPM (Node Package Manager) alata. NPM služi kao programski okvir za Node.js razvoj projekata s velikom količinom raznih dodataka za razne Node.js aplikacije. Ideja NPM nadogradnje je javna dostupnost dodatnih komponenti i resursa koji su dostupni kroz jednostavnu instalaciju. NPM funkcionira na način da stvori jednostavan web poslužitelj iz datoteke koja se nalazi u projektu. Može se koristiti u nizu primjera: poslužuje GUI¹⁰ za preglednike, modificira package.json, pokreće naredbe s klijentske strane, prebacuje konzolu do klijentske strane itd. Dodataka ima zaista mnogo, a za potrebe aplikacije e-Tlak kao glavne valja istaknuti Express.js i Mongoose.js. Jednostavnije rečeno, oni predstavljaju middleware, tj. međuprograme koji djeluju između aplikacije i mreže.

Dakle, u slučaju API-ja i baze podataka aplikacije e-Tlak, nakon instalacije Node.js-a, inicirao se projekt uz pomoć NPM-a. Kreirala se mapa, u njoj se otvorio komandno-linijski prozor i pokrenula se naredba „npm init“. Ispunila su se tražena polja i time se

¹⁰ engl. Graphical User Interface – grafičko korisničko sučelje

stvorio Node.js projekt čije su postavke spremljene u datoteku package.json. U njoj je navedena početna JavaScript datoteka koja se izvodi prilikom pokretanja aplikacije te skripta koja je pokreće. Kao IDE¹¹ je korišten WebStorm, izvođača softvera JetBrains, koji predstavlja moderan način JavaScript razvoja. Potom su se instalirali potrebni alati. Njihova instalacija vrši se uz naredbu „npm install“ koja ih nakon instalacije raspakira u mapu node_modules. Stavka „dependencies“ u datoteci package.json sadrži popis svih instaliranih dodataka vezanih za aplikaciju e-Tlak. Instalirani Express.js predstavlja web radno okruženje i standard za većinu Node.js aplikacija danas. On pokreće back-end i poslužuje web aplikaciju (Express, 2018). Odgovarajuće naredbe vezane za Express.js nalaze se u datoteci app.js. Tu se nalazi temeljna ruta aplikacije e-Tlak, ruta vezana za korisnike i ruta vezana za unose. Točnije, Express.js omogućuje slušanje dolaznih zahtjeva na definiranom portu ovisno o metodi zahtjeva (GET, POST, PUT, DELETE¹²). Mapa routes sadrži dvije datoteke zadužene za baratanje rutama – entries.js i users.js. Uglavnom su vezane za čitanje svih unosa, dodavanje i brisanje unosa, izmjene i sl. Drugi glavni dodatak je Mongoose.js koji predstavlja objektni model podataka za rad s MongoDB bazom podataka. Omogućuje čitanje, pisanje, spajanje i izmjenu podataka u bazi na lakši način (Mongoose, 2018). MongoDB je NoSQL¹³ baza podataka u koju se spremaju podaci, u ovom slučaju, korisnici i njihovi unosi (podaci o krvnom tlaku, pulsu, tjelesnoj masi i vremenska oznaka). Mapa models sadrži tri datoteke – index.js, user.js i entry.js. U datoteci index.js nalaze se postavke baze podataka, dok se u ostalim dvjema nalaze nazivi podataka, tipovi podataka, zadane vrijednosti i sl. Također, tu je za korisnike definirano nekoliko pomoćnih funkcija koje se koriste za sigurnu pohranu lozinke prilikom prijave i registracije korisnika. Mapa config sadrži datoteku passport.js koja se odnosi na autorizaciju korisnika i postavke autorizacije. Ovdje je korišten dodatak JWT¹⁴ (JSON Web Token) koji predstavlja dugačak niz znakova u kojima su šifrirani podaci o korisniku. On se mora dodati u API zahtjeve kako bi API znao prepoznati o kojem se korisniku radi, tj. njegov ID i određeno dopuštenje (JSON Web Tokens, 2018). Od ostalih dodataka treba spomenuti i

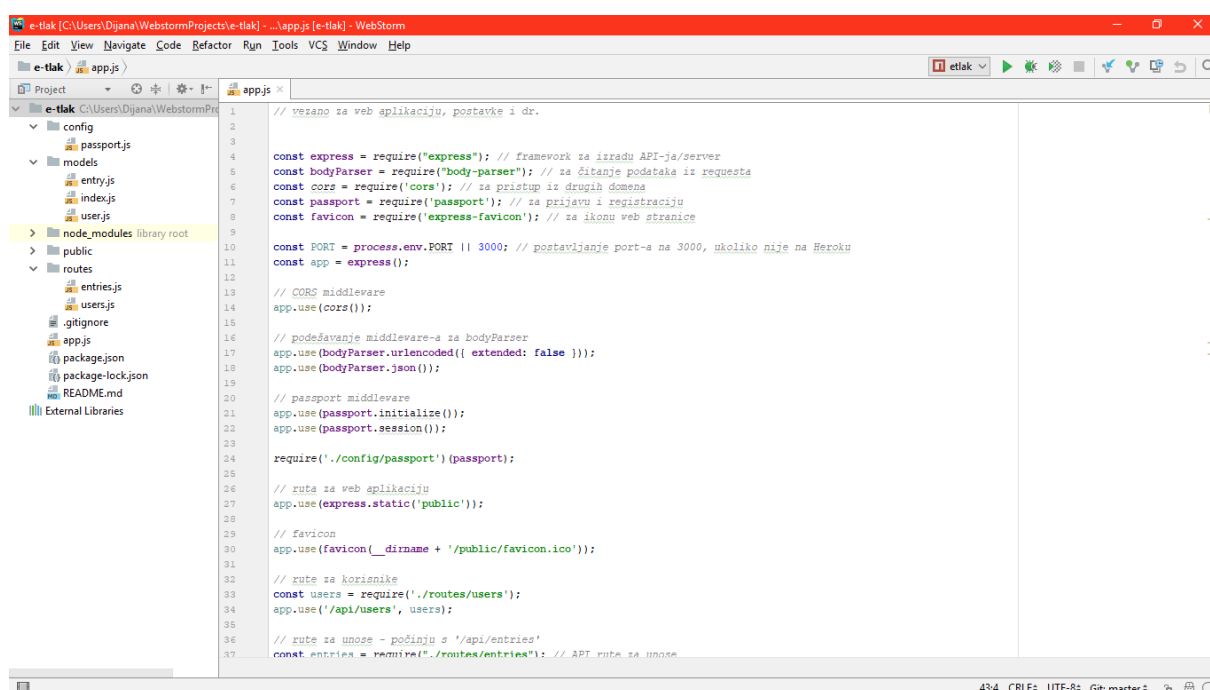
¹¹ engl. Integrated Development Environment – integrirano razvojno okruženje

¹² HTTP metode za RESTful aplikacije

¹³ NoSQL predstavlja skup principa arhitekture baza podataka koji odmiču od centraliziranog, strogo definiranog relacijskog modela i SQL upita. SQL ili Structured Query Language je standardni jezik sustava za upravljanje bazama podataka (Gačić, 2017, p. 55).

¹⁴ engl. JSON Web Token – otvoreni standard koji definira kompaktni i samodovoljni način za sigurno prenošenje informacija između stranaka u JSON ili JavaScript Object Notation formatu za prijenos strukture podataka (Plazonić, 2016, p. 7).

Bcrypt.js koji se odnosi na šifriranje lozinki. Općenito, lozinke se ne smiju spremati u bazu podataka u ljudski čitljivom formatu, pa se unesena lozinka šifrira i kasnije se uspoređuje na isti način, tj. ponovno se šifrira kod prijave korisnika i mora odgovarati spremljenom šifriranom nečitkom obliku (Bcrypt, 2018). Korišten je i CORS dodatak koji omogućuje pristup API-ju iz vanjskih izvora (CORS, 2018) čime je ostvarena komunikacija s mobilnom aplikacijom e-Tlak. Mapa public sadrži web aplikaciju e-Tlak, a njoj se pristupa korištenjem korijenske rute aplikacije. U nastavku je prikazan primjer programskog koda u WebStormu.



```
1 // vezano za web aplikaciju, postavke i dr.
2
3
4 const express = require("express"); // framework za izradu API-ja/server
5 const bodyParser = require("body-parser"); // za čitanje podataka iz requesta
6 const cors = require('cors'); // za pristup iz drugih domena
7 const passport = require('passport'); // za prijavu i registraciju
8 const favicon = require('express-favicon'); // za ikonu web stranice
9
10 const PORT = process.env.PORT || 3000; // postavljanje port-a na 3000, ukoliko nije na Heroku
11 const app = express();
12
13 // CORS middleware
14 app.use(cors());
15
16 // podešavanje middleware-a za bodyParser
17 app.use(bodyParser.urlencoded({ extended: false }));
18 app.use(bodyParser.json());
19
20 // passport middleware
21 app.use(passport.initialize());
22 app.use(passport.session());
23
24 require('./config/passport')(passport);
25
26 // ruta za web aplikaciju
27 app.use(express.static('public'));
28
29 // favicon
30 app.use(favicon(__dirname + '/public/favicon.ico'));
31
32 // rute za korisnike
33 const users = require('./routes/users');
34 app.use('/api/users', users);
35
36 // ruta za unos - počinju s '/api/entries'
37 const entries = require('./routes/entries'); // API ruta za unos
```

Slika 6. Primjer programskog koda back-enda aplikacije e-Tlak u IDE-u WebStorm
(Izvor: autorica)

Višestruka mjerenja i uspoređivanja s drugim sličnim sustavima kao npr. Apache+PHP¹⁵, pokazala su da Node.js uz utrošak veće količine resursa, tj. radne memorije i procesora, u kraćem vremenskom periodu isporučuje traženi sadržaj (Penezić, 2013).

¹⁵ engl. Hypertext Preprocessor

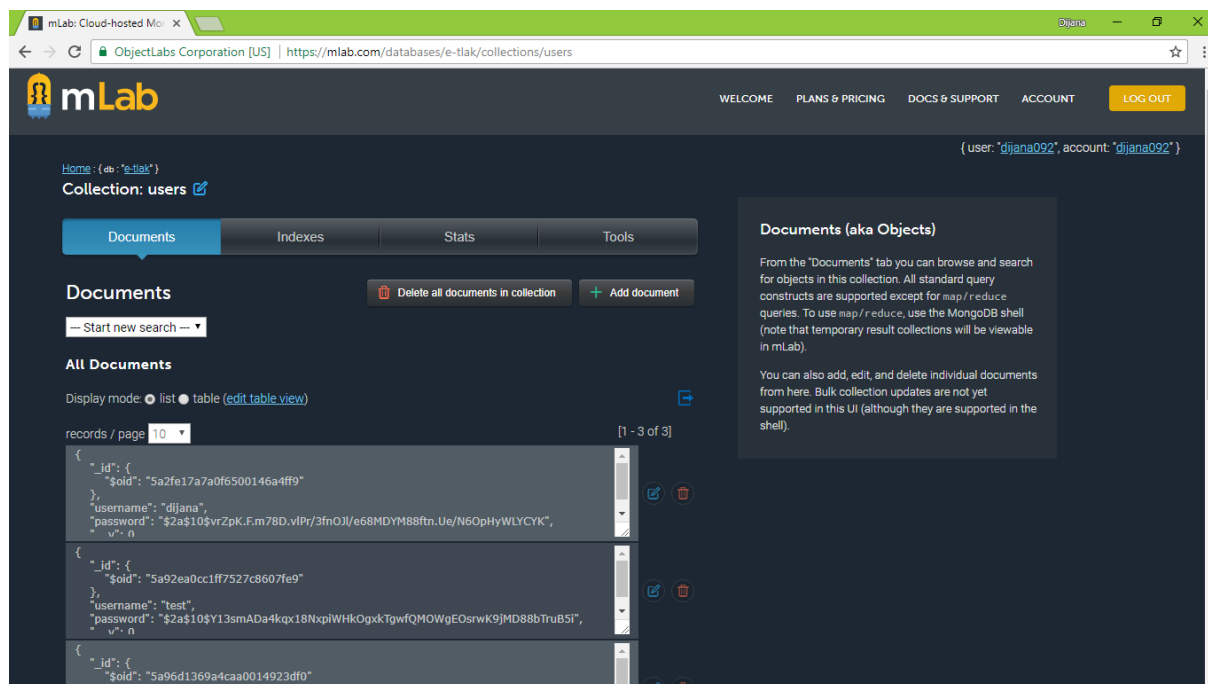
2.2.2. Platforma Heroku i baza podataka mLab

Kako bi back-end dijelu bilo moguće pristupiti putem mobilnog uređaja neovisno o lokaciji, potreban je hosting na stalno pokrenutom poslužitelju. Za potrebe aplikacije e-Tlak, idealan izbor bio je Heroku. To je platforma kao usluga¹⁶ koja omogućava razvoj, pokretanje i upravljanje aplikacijom u oblaku. Heroku podržava sve moderne programske jezike i tehnologije koje su aktualne u današnje vrijeme. Treba istaknuti i da Heroku omogućuje i povezivanje s GitHub platformom za razvoj softvera te izravno git spajanje. Inače, Heroku aplikacije kojima nije pristupljeno 30 minuta, gase se zbog ograničenja od 500 sati mjesečnog hostinga (Heroku, 2018). Iz tog razloga je prvi pristup aplikaciji malo usporen jer se aplikacija treba pokrenuti od nule. Kao glavno što je prevagnulo kao odabir tehnologije Heroku je to što na istoj Heroku poveznici može biti i API i web aplikacija.

Baza podataka se nalazi na udaljenom poslužitelju, a za to je odabran mLab koji predstavlja besplatnu i vodeću bazu podataka kao uslugu¹⁷ vezano za MongoDB (mLab, 2018). Kreiranje i korištenje mLab-a je brzo i jednostavno. U nastavku je prikaz kolekcije vezane za e-Tlak korisnike u mLab-u.

¹⁶ engl. Platform as a Service, PaaS

¹⁷ engl. Database as a Service, DBaaS



Slika 7. Primjer kolekcije baze podataka aplikacije e-Tlak u mLab bazi podataka
(Izvor: autorica)

3. MOBILNA APLIKACIJA E-TLAK

Mobilna aplikacija e-Tlak omogućuje isključivo bilježenje, pohranu i praćenje krvnog tlaka i pohranu tjelesne mase na pametnom telefonu. Razradom funkcionalnosti i uz specifičan odabir tehnologije pronašao se način za maksimalno pojednostavljivanje unosa navedenih podataka, povezivanje na API te je implementirano pregledno mobilno korisničko sučelje vezano za front-end dio aplikacije.

3.1. Razrada funkcionalnosti

3.1.1. *Korisnički scenariji*

Prije same izrade mobilne aplikacije e-Tlak, opisane su i grupirane njezine funkcionalnosti. Napravljen je dijagram slučajeva korištenja¹⁸ koji se odnosi na cijeli sustav mobilne aplikacije, odnosno njegovo ponašanje. Prikazani su dijelovi sustava, korisnik kao vanjski sudionik, baza podataka izvan sustava te veze. Važno je shvatiti način na koji mobilna aplikacija komunicira s korisnikom kao i ostalim sustavima, tj. bazom podataka. Svi dijagrami koji slijede izrađeni su u LucidChart-u, jednostavnom i besplatnom online alatu za izradu dijagrama softvera i vizualnih rješenja.

Mogući korisnički scenariji mobilne aplikacije e-Tlak su:

- registracija korisnika,
- prijava korisnika,
- unos krvnog tlaka (unos sistoličkog tlaka, unos dijastoličkog tlaka i unos pulsa),
- unos tjelesne mase,
- pregled dijagnoze,
- pregled povijesti unosa (pregled sistoličkog tlaka, pregled dijastoličkog tlaka, pregled pulsa, pregled mase i pregled vremena unosa) te
- brisanje unosa.

U nastavku je prikazan dijagram slučajeva korištenja s navedenim korisničkim scenarijima.

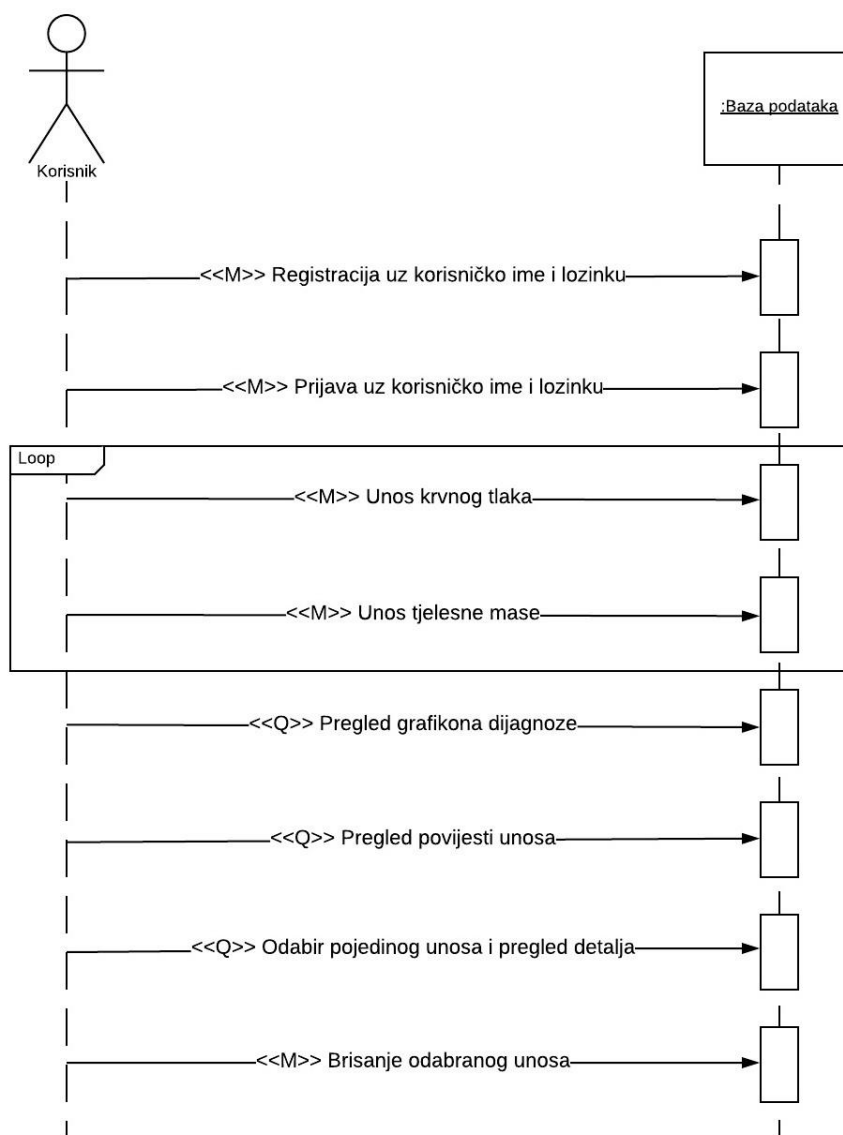
¹⁸ engl. Use Case diagram

kojim se prethodno uspješno registrirao. Nakon toga korisnik je stalno prijavljen u sustav. Kada je korisnik prvi put u mobilnoj aplikaciji, odmah slijedi unos vrijednosti krvnog tlaka i odmah nakon njega i u unos tjelesne mase. Ta dva scenarija inače predstavljaju petlju, s obzirom na to da korisnik stalno unosi svoj krvni tlak i promjenu tjelesne mase. Korisnik ima mogućnost pregledati grafikone dijagnoze. Kao posljednji glavni korisnički scenarij, tu je pregled povijesti unosa gdje korisnik može odabrati pojedini unos i pregledati sve detalje unosa, a zatim mu se nudi i brisanje odabranog unosa ako on to želi.

Dakle, funkcionalnosti za ulogu korisnika u mobilnoj aplikaciji su slijedeći:

- unos, ažuriranje, brisanje i pregled vrijednosti krvnog tlaka,
- unos, ažuriranje brisanje i pregled vrijednosti tjelesne mase,
- pregled grafičkih prikaza i statistika vezanih za dijagnozu te
- pregled i brisanje povijesti unosa krvnog tlaka.

Baza podataka nalazi se izvan mobilne aplikacije i zaprima sve podatke vezane za registraciju i prijavu korisnika, bilježi svaki korisnikov unos vrijednosti krvnog tlaka i tjelesne mase, isporučuje grafikone vezane za pregled dijagnoze i podatke vezane za pregled povijesti unosa te zaprima svaki zahtjev za pregledom pojedinog unosa i detalja kao i za brisanjem unosa od strane korisnika. U nastavku slijedi prikaz sekvencijskog dijagrama.



Slika 9. Sekvencijski dijagram mobilne aplikacije e-Tlak
(Izvor: autorica)

3.1.2. Mobilna aplikacija Dnevnik krvnog tlaka

Prva verzija aplikacije za praćenje krvnog tlaka bila je mobilna aplikacija pod nazivom „Dnevnik krvnog tlaka“. Funkcionalnosti mobilne aplikacije bile su minimalne te je dizajn bio drugačiji. S obzirom na to da je mobilna aplikacija e-Tlak nadogradnja navedene aplikacije, valja napomenuti da su njezine funkcionalnosti poslužile za razradu detaljnijih funkcionalnosti mobilne aplikacije e-Tlak, dok je izgled zaslona poslužio kao smjernica i jedna vrsta prototipa za redizajn.

Funkcionalnosti su se odnosile prvenstveno na pohranu krvnog tlaka, praćenje krvnog tlaka i izvještavanje korisnika o krvnom tlaku na pametnom telefonu. Mobilna

aplikacija nije imala dio vezan za registraciju i prijavu korisnika. Prilikom pokretanja mobilne aplikacije otvarao se početni zaslon koji je u prvom dijelu obuhvaćao informacije o vremenu zadnjeg mjerenja krvnog tlaka, njegovim vrijednostima sistoličkog i dijastoličkog tlaka te pulsa. Drugi dio zaslona sadržavao je tablicu s podacima o posljednja tri unosa, tj. vremenu, tlaku i pulsu. Sljedeći zaslon, odnosno zaslon vezan za unos krvnog tlaka, omogućavao je jednostavan unos ključnih vrijednosti krvnog tlaka u obliku sistoličkog i dijastoličkog krvnog tlaka te pulsa kroz pametnu tipkovnicu koja je omogućavala lakši unos i brisanje pogrešnog unosa. Pohrana svih napravljenih unosa krvnog tlaka slala se na lokalnu bazu podataka mobilnog uređaja. Zaslon vezan za dijagnozu izvještavao je korisnika o najzastupljenijoj dijagnozi kroz tortni grafikon koji je sadržavao točan postotak za svaku prisutnu dijagnozu. Zaslon vezan za povijest, kao posljednji zaslon, sadržavao je dinamičku kronološku listu evidentiranih vrijednosti krvnog tlaka te je nudio mogućnost čitanja svakog napravljenog unosa kao i njegovo brisanje. Za dizajn mobilne aplikacije korištena je svijetla i srednja nijanse plave boje uz bijelu boju.

Za izradu mobilne aplikacije bio je korišten programski jezik Java i operacijski sustav Android. Detaljnije, arhitekturu je činio sloj za pohranu i pristup podacima (baza podataka SQLite), sloj za validaciju korisničkih unosa u aplikaciju (klasa Validacija.java), prezentacijski sloj za prikaz podataka (klasa ListFragment.java), sloj za grafičku dijagnozu zdravlja korisnika (biblioteka MPAndroidChart) i sloj za poslovne logike za objektno orijentirani pristup (klasa KrvniTlak.java).

Mobilna aplikacija Dnevnik krvnog tlaka zahtijevala je redizajn s obzirom na malu veličinu fonta i elemenata zaslona. Nedostajala je opcija za unos dodatnih podataka, poput tjelesne mase. S obzirom na to da je postojala samo mobilna aplikacija, došlo do potrebe da se izradi i web aplikacija za detaljniju vizualizaciju podataka vezanih za krvni tlak te je bilo poželjno napraviti bazu podataka izvan lokalne memorije mobilnog uređaja. U nastavku je prikaz svih prethodno opisanih zaslona mobilne aplikacije Dnevnik krvnog tlaka.



Slika 10. Zasloni mobilne aplikacije Dnevnik krvnog tlaka
(Izvor: autorica)

3.2. Implementacija i korištene tehnologije

Nakon razrade funkcionalnosti vezanih za mobilnu aplikaciju e-Tlak, uslijedila je i sama njezina izrada. Mobilnoj aplikaciji e-Tlak se pristupa uz pomoć pametnog telefona, instaliranjem aplikacije te je za njeno korištenje potreban pristup internetu. Mobilna aplikacija se uz pomoć interneta spaja na API putem kojeg komunicira s udaljenom bazom podataka te uz pomoć kojeg je ostvarena komunikacija s web aplikacijom e-Tlak. Kao glavne tehnologije koje su se koristile za izradu mobilne aplikacije korišten je programski jezik Java i operacijski sustav Android te je nekoliko

važnih Android biblioteka poput biblioteke Volley, MPAndroidChart biblioteke i GraphView biblioteke.

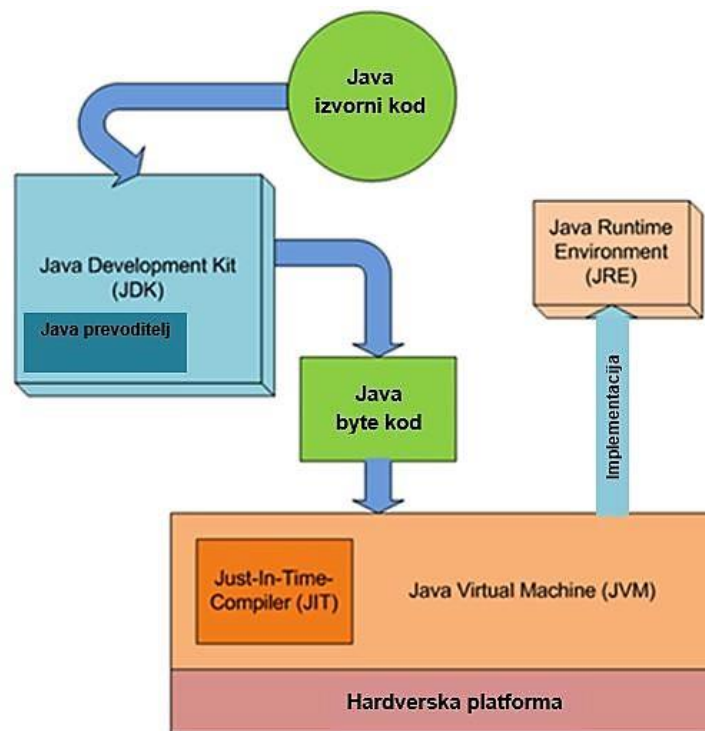
3.2.1. Programski jezik Java

Programski jezik Java izradila je kompanija Sun Microsystems, na inicijaciju Jamesa Goslinga. Rad na samom jeziku započeo je oko 1991. godine kada je bio cilj stvoriti novi prenosivi programski jezik. Prvi put Java je objavljena 1995. godine. Java je sada u vlasništvu kompanije Oracle. Slogan Jave sve govori: „Write Once, Run Everywhere“. Temeljna načela Jave su: jednostavnost, objektna-orijentiranost, robusnost i sigurnost, neovisnost o arhitekturi, portabilnost te brzina (TutorialsPoint, 2014, pp. 15-16). Java se temelji na klasama, odnosno objektima te ima svoju specifičnu sintaksu te uključuje razna sučelja, kolekcije, pakete i biblioteke.

Što se tiče arhitekture Jave, važnu ulogu imaju:

- Java Development Kit (JDK),
- Java Runtime Environment (JRE) i
- Java Virtual Machine (JVM).

JDK omogućuje razvijanje i pokretanje Java aplikacija te uključuje JRE uz alate za prevođenje i pakiranje. JRE je okruženje za pokretanje Java aplikacija što uključuje JVM. JVM predstavlja Java virtualni stroj ili virtualno računalo koje omogućuje da se programi pisani u Javi prevode u strojni jezik. Taj strojni jezik zove se Java byte kod, a do njega se dolazi uz pomoć JIT prevoditelja. Java byte kod može biti izvršen na bilo kojem računalu na kojem je instaliran Java prevoditelj. Dakle, kod pisanja Java aplikacija postupak obuhvaća prevođenje Java izvornog koda u byte kod i izvršavanje istog prevođenjem unutar JVM-a. Prevedena aplikacija je prenosiva između platformi bez potrebe za ponovnim prevođenjem (Savitch, 2016, pp. 33-60). Slika u nastavku prikazuje odnose i funkcioniranje JDK-a, JVM-a i JRE-a.



Slika 11. Arhitektura programskog jezika Java
(Izvor: Adarsh, 2015)

Prema globalnim statistikama, Java je jedan od najkorištenijih i najpopularnijih programskih jezika. Široko je rasprostranjen i dosljedno se koristi tamo gdje prevladava brzina razvoja programskog sustava nad zahtjevima do brzine rada programa. Milijuni programera rade u Javi te je vrijednost poslova obavljenih upotrebom Java tehnologija velika. Uz Javu se izrađuju programi koji mogu izvršavati više zadataka simultano, razne interaktivne aplikacije, apleti ili male Java aplikacije, dinamičke web stranice, složene web aplikacije te ona predstavlja osnovni jezik za programiranje aplikacija Googleovog sustava Android (Statistics Times, 2017).

3.2.2. Operacijski sustav Android

Android predstavlja svjetski vodeću i najpopularniju mobilnu platformu. Stotine milijuna mobilnih uređaja u više od 190 zemalja širom svijeta zasniva se na Androidu, a brojke sve više rastu. Uz Linux zajednicu i više od 300 partnera, Android je postao najbrže rastući mobilni operacijski sustav (Android Developers, 2018). Android Inc. osnovan je 2003. godine, a osnovali su ga Andy Rubin, Rich Miner, Nick Sears i

Chris White s ciljem razvijanja programa za pametne mobilne uređaje uzimajući u obzir korisničke postavke i lokaciju korisnika. Nakon nekog vremena Google kupuje Android te spomenuti osnivači i programeri dobivaju pojačanje Googleovih programera. Android je zamišljen kao projekt otvorenog koda te je od 2008. godine dostupan cjeloviti kod pod Apache licencom. Ubrzani porast broja korisnika i podržanih uređaja Android postiže konstantnim izdavanjem novijih verzija operacijskog sustava koje podrazumijevaju poboljšanja i novitete (DiMarzio, 2017, pp. 2-26). Programerima omogućuje inovativnu izgradnju naprednih aplikacija koje koriste najnovije mobilne tehnologije.

Platforma podrazumijeva operacijski sustav i razvojnu okolinu. Android Software Development Kit (SDK) predstavlja skupinu alata koji omogućavaju stvaranje aplikacija za Android operacijski sustav i pružaju podršku za razvoj, testiranje, pronalaženje i uklanjanje pogrešaka. Za pohranu podataka koristi se SQLite relacijski DBMS²⁰ napisan u programskom jeziku C. Iako je većina aplikacija pisana u programskom jeziku Java, na Android uređajima ne postoji JVM, pa tako nije moguće izvršavati Java byte kod. Za pokretanje Java aplikacija Android koristi virtualni stroj Dalvik.

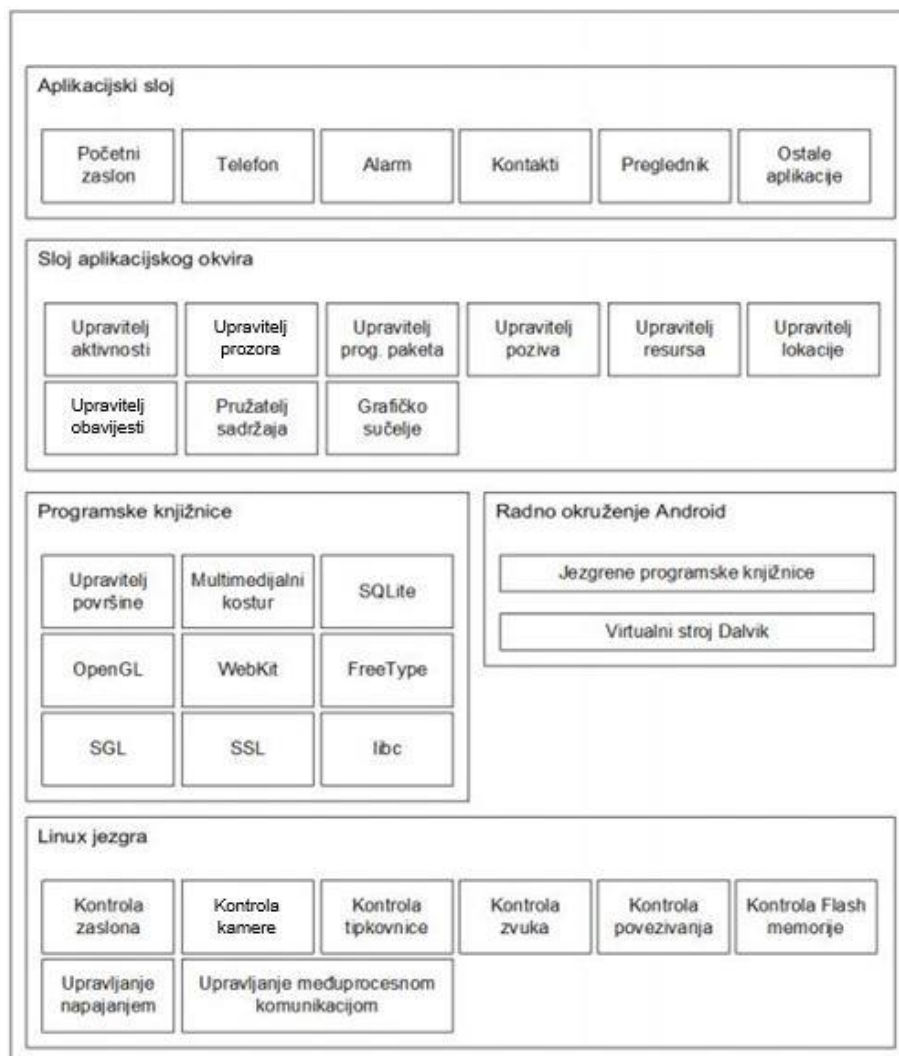
Arhitekturu operacijskog sustava Android čini:

- Linux jezgra,
- programske knjižnice,
- radno okruženje Android,
- sloj aplikacijskog okvira i
- aplikacijski sloj.

Linux jezgra obuhvaća upravljačke mehanizme pomoću kojih se omogućuje zaštićeno izvođenje naredbi unutar kontroliranih procesa. Što se tiče programskih knjižnica ili biblioteka, njihova je namjena pružanje specifičnih usluga ostalim dijelovima sustava na višim slojevima. Radno okruženje Android je smješteno u istom sloju s bibliotekama i nudi set temeljnih biblioteka koje omogućuju programerima kreiranje aplikacija koristeći programski jezik Java. Također, uključuje i navedeni virtualni stroj Dalvik. Sloj aplikacijskog okvira čine komponente upravitelja usluga i pružatelja sadržaja. Aplikacijski sloj obuhvaća instalirane aplikacije koje pružaju

²⁰ engl. Database Management System – sustav za upravljanje bazom podataka

osnovne usluge za korisnike, a to su primjerice: kalkulator, kamera, kontakti, razni preglednici i sl (Android Developers, 2018). Opisana arhitektura Androida prikazana je na slici.



Slika 12. Arhitektura operacijskog sustava Android
(Izvor: Popić, 2015, p. 4)

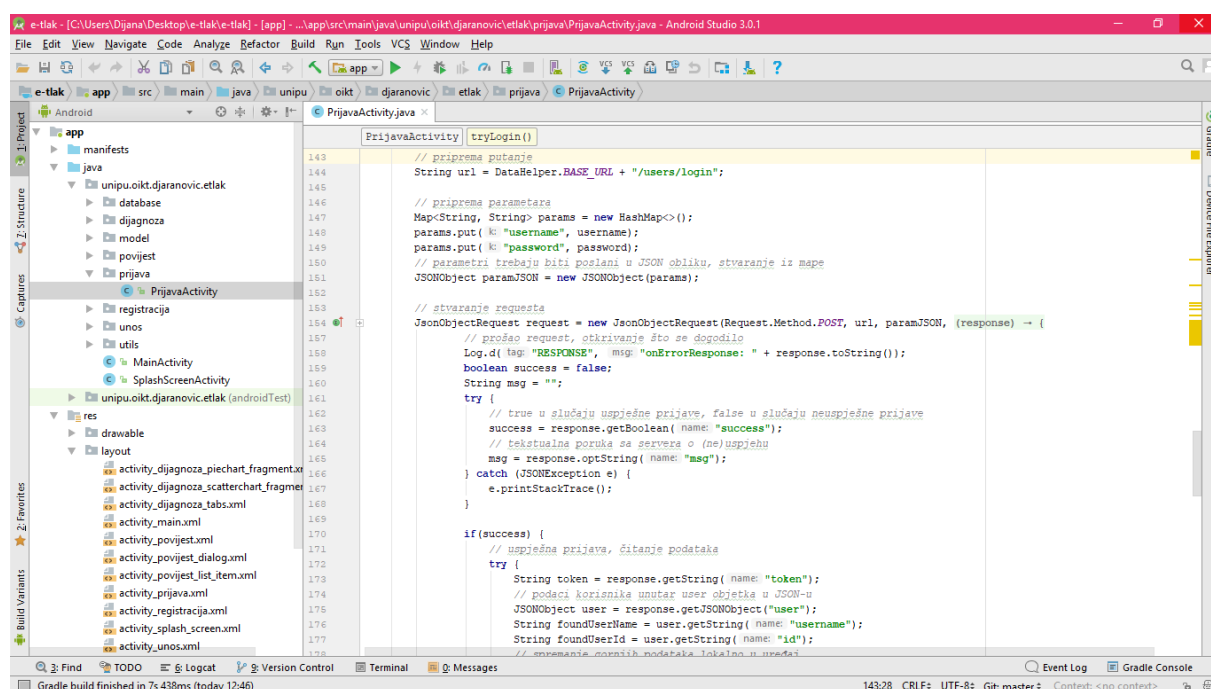
3.2.3. IDE Android Studio i biblioteka Volley

Vrlo popularan IDE je Android Studio u kojem je rađena mobilna aplikacija e-Tlak. Temeljen je na razvojnom okruženju IntelliJ IDEA, u Google-ovom je vlasništvu te nudi mnogo opcija za razvoj mobilnih aplikacija za operacijski sustav Android.

Android Studio, odnosno projekt svake mobilne aplikacije sadrži važnu datoteku `AndroidManifest.xml` koja se nalazi u glavnom direktoriju paketa i sadrži ključne informacije o mobilnoj aplikaciji koje Android sustav mora imati da bi mogao pokrenuti bilo koji dio koda (Phillips et al., 2015, pp. 17-20). Što se tiče mobilne aplikacije e-Tlak, kako bi se mogla povezati s mrežom, u `AndroidManifest.xml` je dodano `android.permission.INTERNET` dopuštenje koje se odnosi na važnu biblioteku vezanu za olakšano i brzo slanje API zahtjeva - Volley. Ona funkcionira tako da se stvori `RequestQueue` kao red Volley zahtjeva i u njega se dodaju zahtjevi koji se stvaraju kao objekti. Najčešće se radi o objektima kao što su `JsonObjectRequest` i `JSONArrayRequest`, ovisno o očekivanim povratnim podacima (Android Developers, 2018). Dakle, potrebno je prenijeti podatke na poslužitelj u JSON obliku što je napravljeno pomoću Java Map sučelja (ključ, vrijednost) i ubačeno u konstruktor objekta. U ovom slučaju, Volley pomaže kod prijave i registracije, provjere ima li unosa kod prvog pokretanja mobilne aplikacije, unosa krvnog tlaka i tjelesne mase, postavljanja podataka na glavnoj stranici, dijagnozi i povijesti unosa te kod brisanja unosa povijesti.

Mobilnu aplikaciju e-Tlak čine aktivnosti, tj. Java klase vezane za: bazu podataka (`DataHelper.java`), glavni dio (`MainActivity.java`), prijavu (`PrijavaActivity.java`), registraciju (`RegistracijaActivity.java`), unos (`UnosActivity.java`, `UnosMasaActivity.java` i `ValidacijaUnosa.java`), dijagnozu (`DijagnozaTabsActivity.java`, `PieChartFragment.java`, `ScatterChartFragment.java` i `DecimalRemover.java`), povijest (`PovijestActivity.java`), model podataka (`KrvniTlak.java`) i pomoćne klase (`ActivityHelper.java` i `SplashScreenActivity.java`). Radi se uglavnom o osnovnim fokusiranim stvarima koje korisnik može raditi, poput unosa, pregleda dijagnoze, povijesti i dr. Ovdje treba posebno istaknuti `DataHelper.java` klasu koja sadrži temeljnu API putanju, olakšava čitanje i spremanje podataka o korisniku, odjavu te sadrži pomoćne funkcije za broj unosa vezano za pametnu navigaciju prilikom prvog pokretanja, funkciju vezanu za format datuma i dr. Najvažnija zadaća ove klase je čitanje tokena koji se dobije prilikom uspješne prijave korisnika. Taj se token pridodaje zahtjevima za čitanje i brisanje unosa i po njemu se zna o kojem se korisniku radi. Zvuk nakon uspješnog unosa omogućila je `MediaPlayer` klasa za reproduciranje zvuka. Što se tiče dijagnoze, korišten je `ViewPager` i `Fragmenti` kao poseban način prikaza i ponašanja sučelja. Tu su

korištene MPAAndroidChart i GraphView biblioteke koje podržavaju razne vrste grafikona kao i njihovo uređivanje i animiranje. Kod prikaza povijesti unosa korišten je ListView koji omogućuje prikaz u obliku liste. Korisničko sučelje mobilne aplikacije što uključuje deklarirane dijelove aplikacije i cjelokupni dizajn izgleda i ikone definirano je u XML²¹-u. U nastavku je prikazan primjer programskog koda u Android Studio.



Slika 13. Primjer programskog koda mobilne aplikacije e-Tlak u IDE-u Android Studio
(Izvor: autorica)

Projekt mobilne aplikacije e-Tlak dostupan je na GitHub platformi, na adresi: <https://github.com/dijana092/e-tlak-mobilna>. Općenito, nakon što je mobilna aplikacija gotova, postavlja se na trgovinu aplikacija Google Play kako bi postala dio zajednice i jedne velike baze aplikacija.

3.3. Korisničke upute

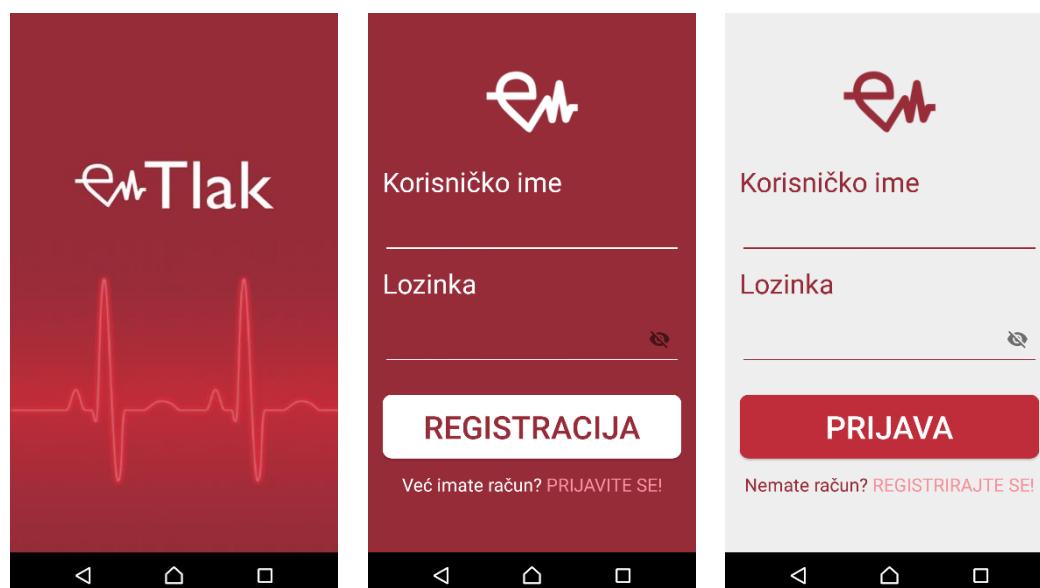
Sljedeća i trenutna verzija mobilne aplikacije za praćenje krvnog tlaka nazvana je “e-Tlak”. U odnosu na prvu verziju, funkcionalnosti mobilne aplikacije e-Tlak su dorađene i proširene te je dizajn promijenjen. Dizajn ove mobilne aplikacije ističe se

²¹ engl. Extensible Markup Language – jezik za označavanje podataka

mekim i pastelnim nijansama crvene boje te bijelom bojom, dok su veličina fonta i elementi zaslona znatno povećani i više prilagođeni starijim korisnicima. Način na koji se mobilna aplikacije e-Tlak koristi prilično je jednostavan.

3.3.1. Registracija i prijava

Prilikom pokretanja mobilne aplikacije, nakon kraćeg prikaza zaslona s logotipom, otvara se zaslon za registraciju te prijavu. Ako račun nije izrađen, potrebna je registracija, dok ako račun već postoji, potrebno je samo prijaviti se. Za to je dovoljno obično korisničko ime i lozinka po želji, ukoliko je dostupno. Ukoliko je tek izvršena registracija, nakon prijave i kod prvog unosa, automatski se otvara zaslon za unos krvnog tlaka, a odmah nakon i za unos tjelesne mase. Inače, otvara se glavni zaslon mobilne aplikacije. Dakle, navigiranje kroz aplikaciju i unos podataka je maksimalno pojednostavljen.

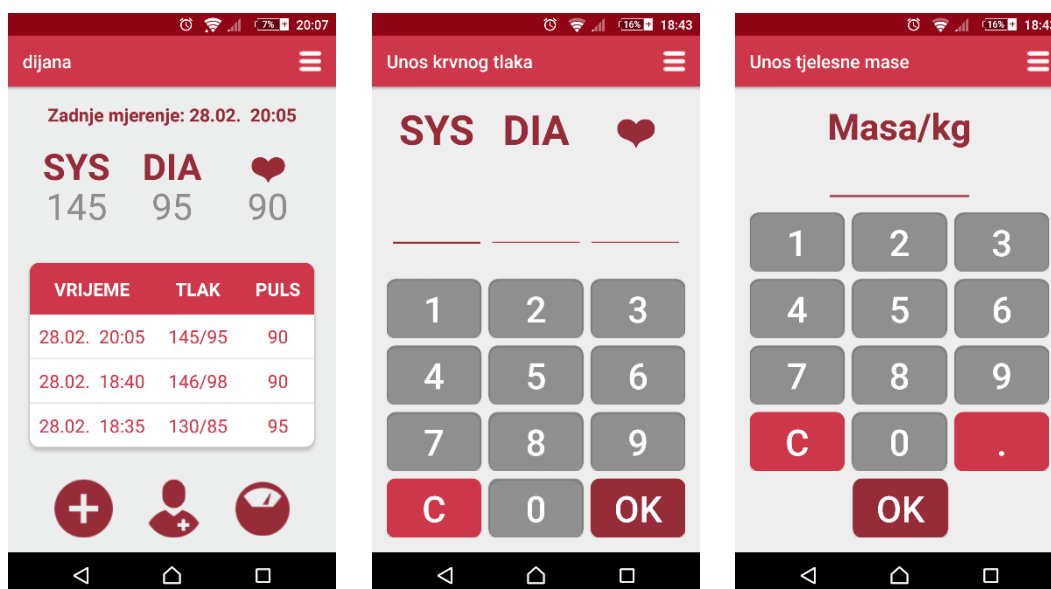


Slika 14. Zaslon s logotipom i zasloni registracije i prijave mobilne aplikacije e-Tlak
(Izvor: autorica)

3.3.2. Unos krvnog tlaka i tjelesne mase

Glavni zaslon mobilne aplikacije je sadržajno isti kao i u prvoj verziji: u prvom dijelu zaslona nalaze se podaci o zadnjem mjerenju krvnog tlaka, a u drugom dijelu

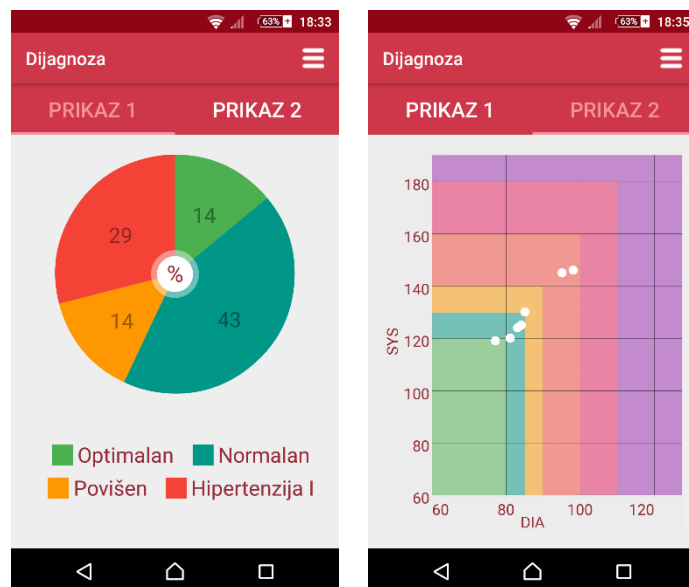
nalazi se tablica s podacima o posljednja tri unosa. Svi gumbi na sljedeće aktivnosti smješteni su na samom dnu zaslona. Zaslون vezan za unos krvnog tlaka redizajniran je drugačijim razmještajem formi za unos te uvećanom pametnom tipkovnicom koja funkcionira na isti način kao i u prvoj verziji mobilne aplikacije, tj. omogućuje olakšan unos i brisanje sistoličke i dijastoličke vrijednosti te pulsa. Dodan je zaslon vezan za unos tjelesne mase. To je nova mogućnost koja nudi unos kilaže što kasnije čini još jednu dodatnu vizualizaciju i pridonosi lakšem donošenju zaključaka oko dijagnoze. Također, nakon svakog unosa krvnog tlaka ili tjelesne mase prisutan je i zvuk kao potvrda uspješnog unosa. Osim dodanog zvuka, tu su i tekstualne obavijesti većeg fonta.



Slika 15. Glavni zaslon i zaslони za unos krvnog tlaka i tjelesne mase mobilne aplikacije e-Tlak
(Izvor: autorica)

3.3.3. Prikaz dijagnoze

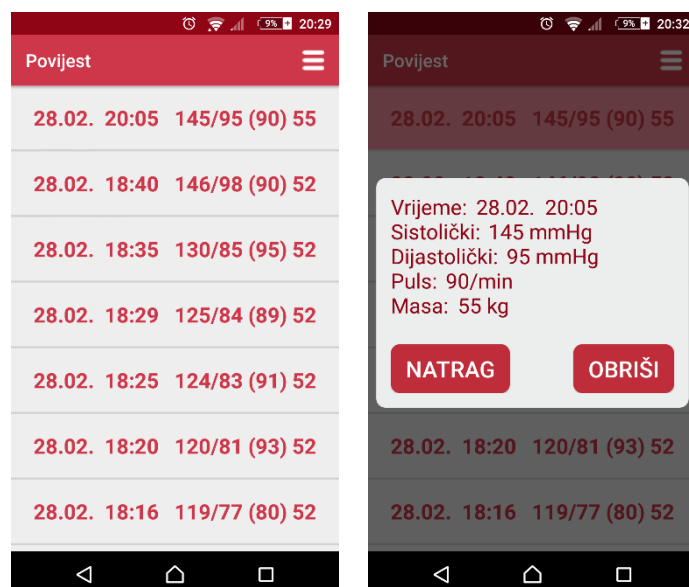
Što se tiče prikaza dijagnoze krvnog tlaka, osim tortnog grafikona, prisutan je i raspršeni grafikon. Oznake i podaci grafikona znatno su uvećani i prilično vidljivi.



Slika 16. Zaslone prikaza dijagnoze mobilne aplikacije e-Tlak
(Izvor: autorica)

3.3.4. Povijest unosa

Zaslon vezan za povijest unosa sadrži listu svih unosa kronološki poredanih. Svaki redak nudi prikazano točno vrijeme unosa, krvni tlak i tjelesnu masu. Odabirom određenog unosa otvara se dijaloški okvir s detaljima unosa te je prisutna mogućnost brisanja.



Slika 17. Zaslone pregleda povijesti unosa mobilne aplikacije e-Tlak
(Izvor: autorica)

4. WEB APLIKACIJA E-TLAK

Web aplikacija e-Tlak služi za pregled raznih vizualizacija i detalja vezanih za krvni tlak kao i tjelesnu masu putem internetskog preglednika na računalu, koji su prethodno uneseni kroz mobilnu aplikaciju e-Tlak. Što se tiče web aplikacije, također su detaljno razrađene i njezine funkcionalnosti, poseban fokus odnosi se na front-end dio te su odabrane određene web tehnologije vezane za implementaciju.

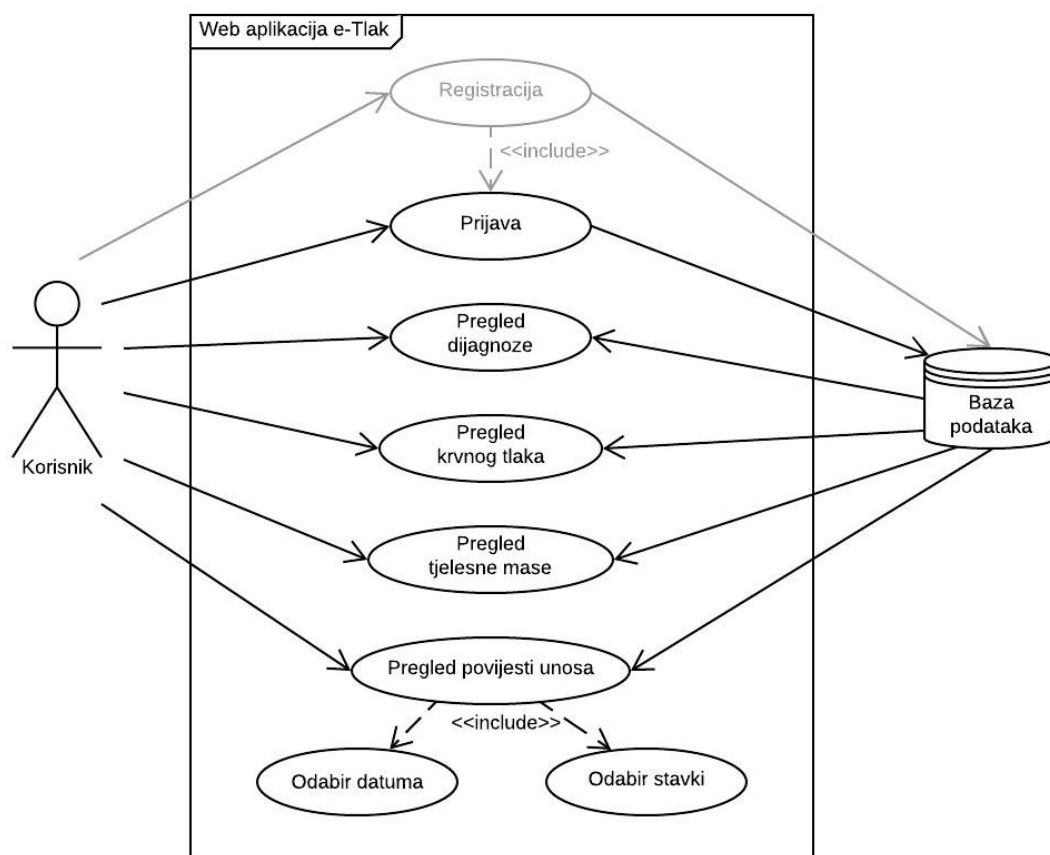
4.1. Razrada funkcionalnosti

4.1.1. Korisnički scenariji

Dijagram slučajeva korištenja web aplikacije e-Tlak odnosi se na cijeli sustav web aplikacije, korisnika kao vanjskog sudionika te komunikaciju s bazom podataka izvan sustava.

Korisnički scenariji web aplikacije e-Tlak koji su mogući, prikazani su na slici u nastavku i tu spadaju:

- registracija korisnika (ako nije obavljena na mobilnoj aplikaciji),
- prijava korisnika,
- pregled dijagnoze,
- pregled krvnog tlaka,
- pregled tjelesne mase te
- pregled povijesti unosa (odabir datuma unosa i odabir stavki prikaza).



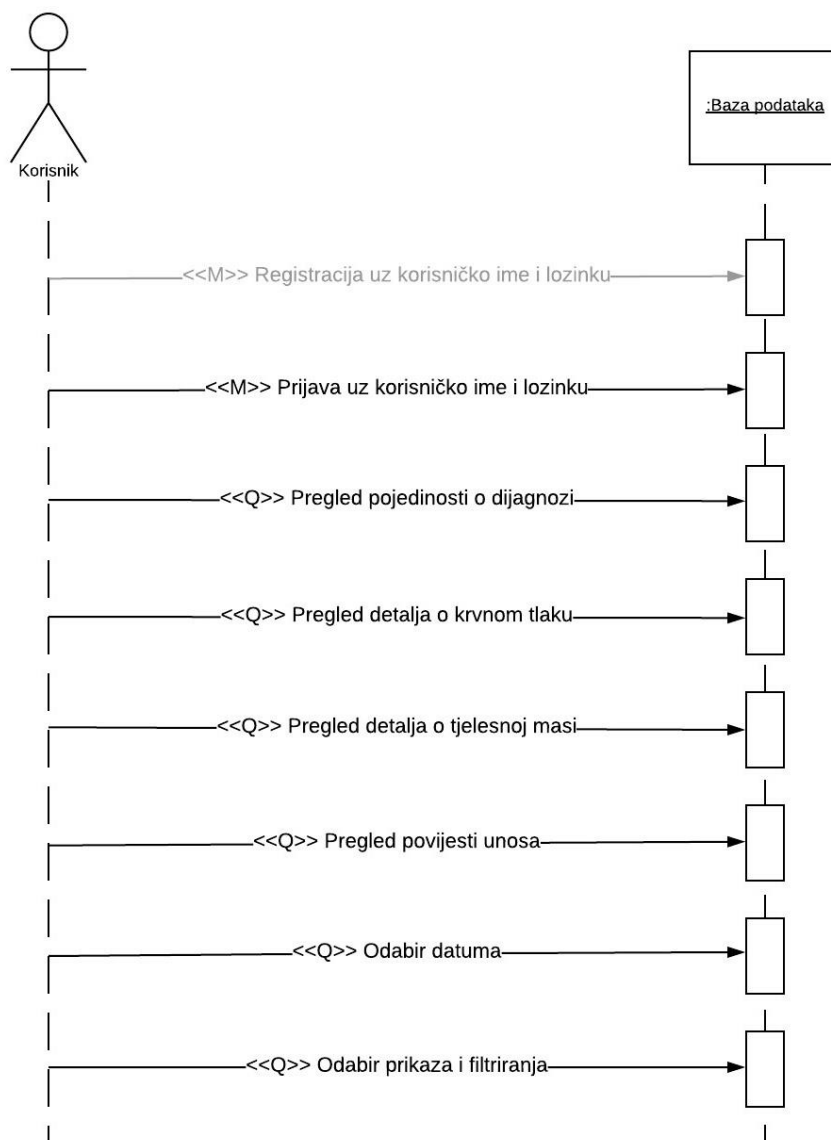
Slika 18. Dijagram slučajeva korištenja web aplikacije e-Tlak
(Izvor: autorica)

Korisnički scenariji web aplikacije e-Tlak detaljnije su razrađeni kroz sekvencijski dijagram. Uglavnom, predviđeno je da se korisnik putem web aplikacije prijavi uz korisničko ime i lozinku kojim to prethodno čini na mobilnoj aplikaciji. Za svaki slučaj, previđena je i opcija registracije ukoliko je odluči učiniti putem web aplikacije. Korisniku se zatim nudi nekoliko glavnih pregleda: pregled dijagnoze sa svim pojedinostima o dijagnozi, pregled krvnog tlaka sa svim detaljima o krvnom tlaku i pregled tjelesne mase sa svim detaljima o tjelesnoj masi. Posljednji pregled je pregled povijesti unosa gdje korisnik može odabrati datum te način prikaza i filtriranja svih unosa krvnog tlaka.

Funkcionalnosti za ulogu korisnika u web aplikaciji e-Tlak su:

- pregled pojedinosti o dijagnozi,
- pregled detalja o krvnom tlaku,
- pregled detalja o tjelesnoj masi te
- odabir, filtriranje i pregled povijesti unosa krvnog tlaka.

Navedenim funkcionalnostima potpomaže baza podataka izvan web aplikacije koja zaprima podatke vezane za prijavu (i registraciju) korisnika te isporučuje sve navedene prikaze koje korisnik želi pregledavati. U nastavku je prikaz sekvencijskog dijagrama vezanog za web aplikaciju.



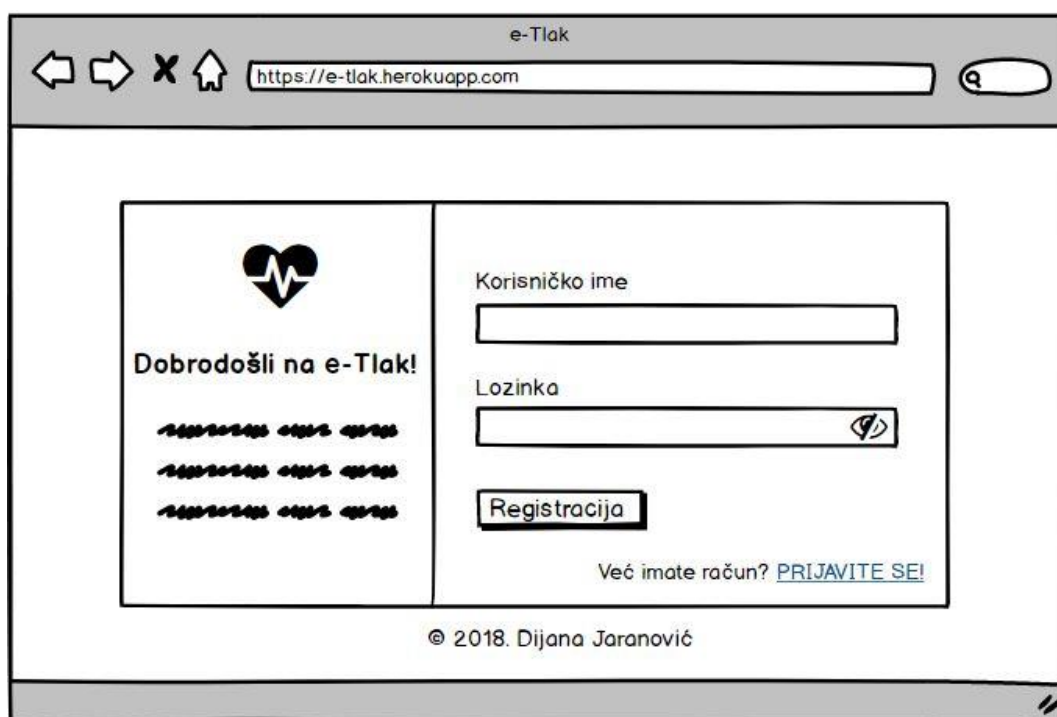
Slika 19. Sekvencijski dijagram web aplikacije e-Tlak
(Izvor: autorica)

4.1.2. Prototip sučelja

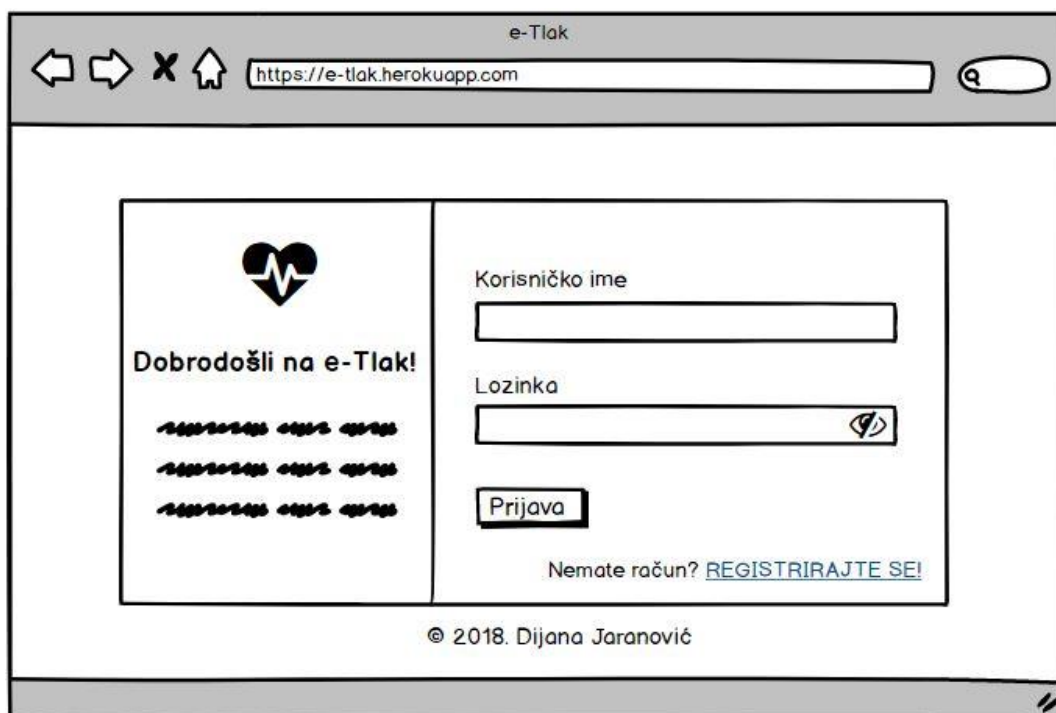
Općenito, prije same izrade web aplikacije poželjno je napraviti prototip sučelja. Prototip služi kao osnova prije izrade prave aplikacije te ne mora u

potpunosti vizualno i tehnički odgovarati gotovoj aplikaciji. On predstavlja orijentir za daljnju izradu aplikacije, tj. doprinosi kasnijem generiranju više ideja ili izbacivanju onih lošijih, kao i otkrivanju boljih rješenja vezanih za aplikaciju. Što se tiče prototipa web aplikacije e-Tlak, izrađen je u alatu Balsamiq, namijenjenom za brzu i jednostavnu izradu skica zaslona aplikacije.

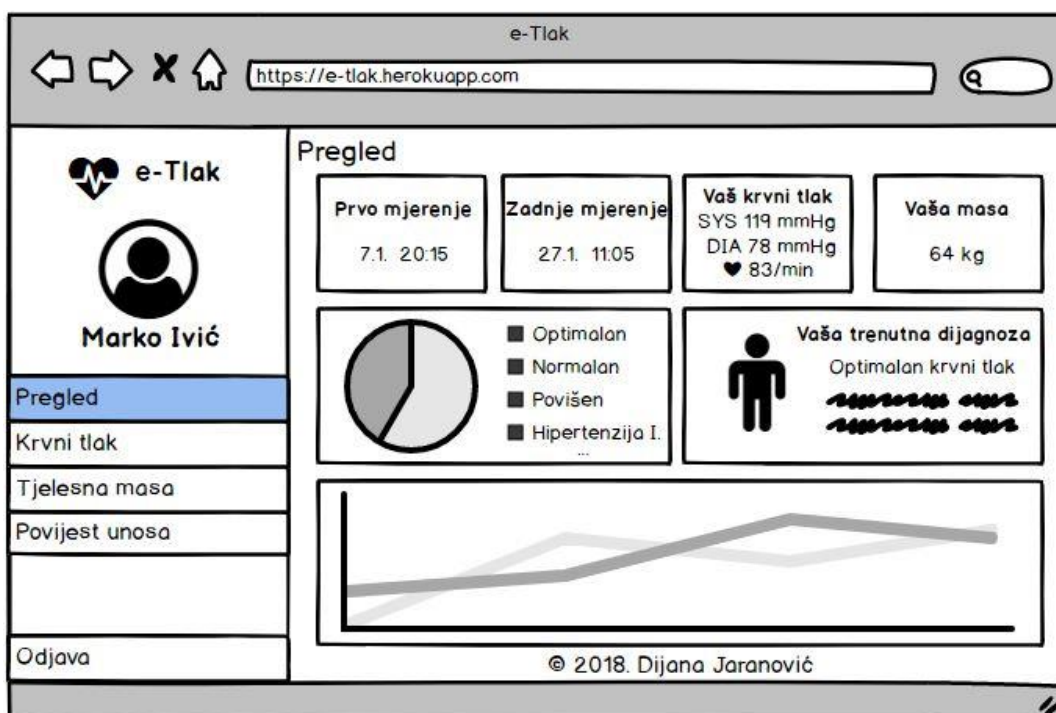
Od samoga početka, zamišljen je početni zaslon vezan za registraciju i prijavu korisnika te glavni dio web aplikacije uz navigaciju s karticom vezanom za početnu vizualizaciju pregleda, karticom vezanom za vizualizaciju krvnog tlaka, karticom vezanom za vizualizaciju tjelesne mase i karticom s povijesti unosa. U nastavku slijede slike skiciranih navedenih zaslona web aplikacije izrađenih u spomenutom alatu.



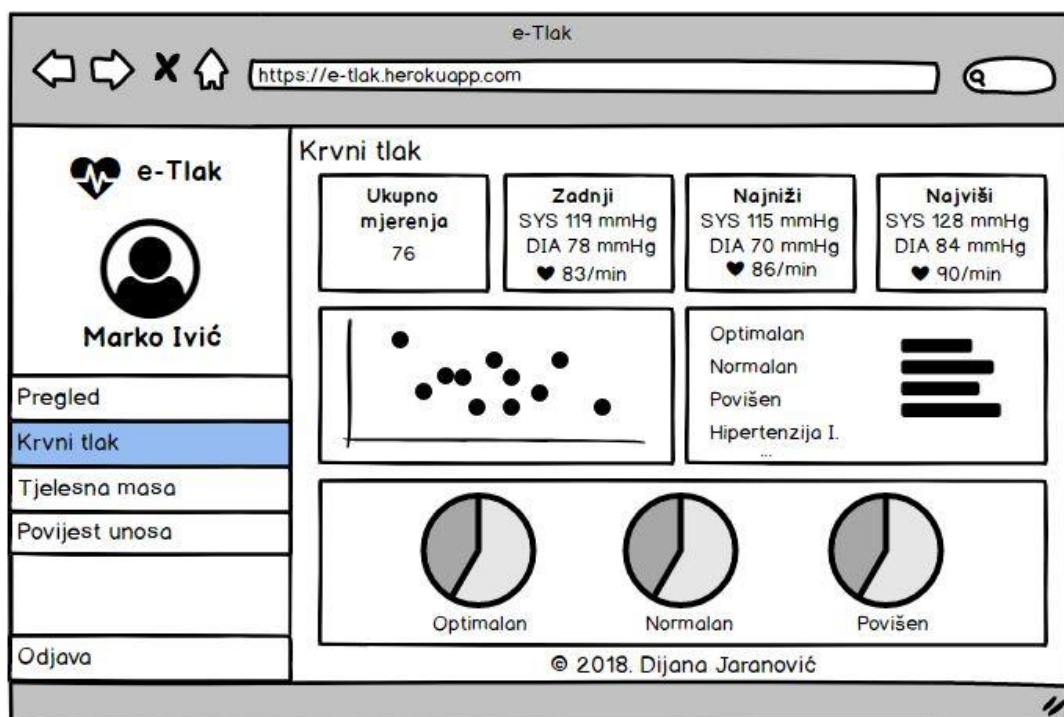
Slika 20. Prototip sučelja web aplikacije e-Tlak: Registracija korisnika
(Izvor: autorica)



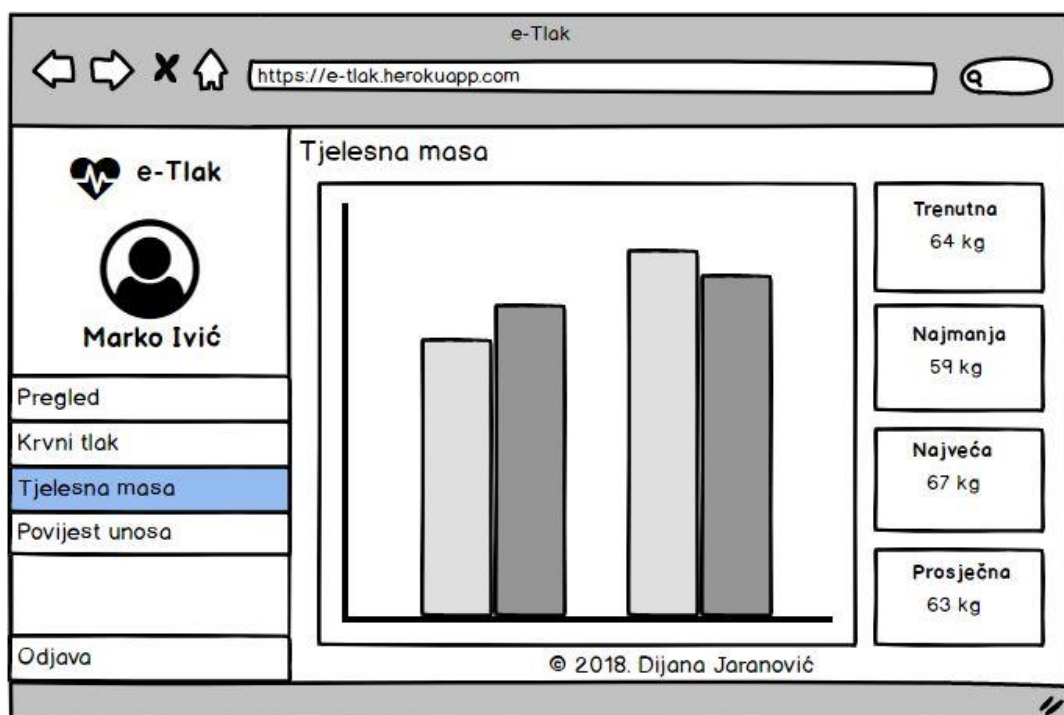
Slika 21. Prototip sučelja web aplikacije e-Tlak: Prijava korisnika
(Izvor: autorica)



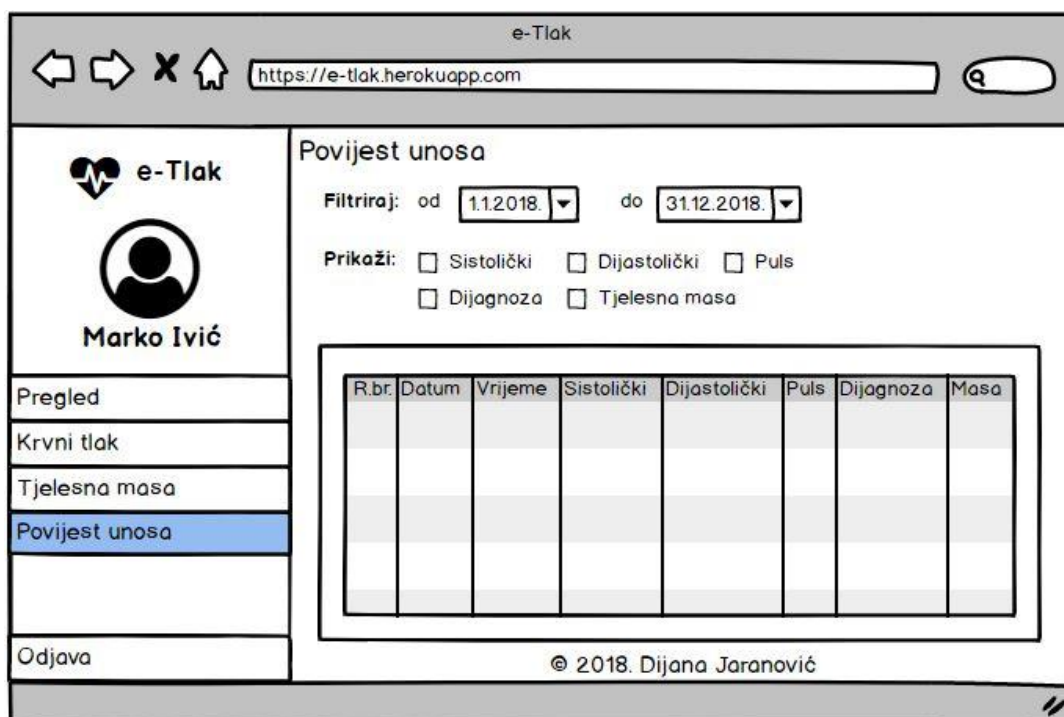
Slika 22. Prototip sučelja web aplikacije e-Tlak: kartica Pregled
(Izvor: autorica)



Slika 23. Prototip sučelja web aplikacije e-Tlak: kartica Krvni tlak
(Izvor: autorica)



Slika 24. Prototip sučelja web aplikacije e-Tlak: kartica Tjelesna masa
(Izvor: autorica)



Slika 25. Prototip sučelja web aplikacije e-Tlak: kartica Povijest unosa
(Izvor: autorica)

4.2. Implementacija i korištene tehnologije

Što se tiče implementacije web aplikacije e-Tlak, u nastavku će se detaljnije pojasniti korišteni jezici JavaScript, HTML i CSS te spomenuti ostale tehnologije i alati kao što su Vue.js, Chart.js, jQuery, Materialize te također Express.js.

4.2.1. JavaScript, HTML i CSS jezici

JavaScript je jednostavan, interpretiran programski jezik koji omogućuje izvršavanje određenih radnji u inače statičnim HTML dokumentima, npr. interakciju s korisnikom, promjenu svojstava preglednikova prozora ili dinamičko stvaranje HTML sadržaja. Zapravo, to je najpopularniji skriptni jezik na Internetu kojeg podržavaju svi poznatiji internetski preglednici. Cilj kreiranja JavaScript jezika bio je dodati interaktivnost HTML stranicama. Službenim početkom razvoja JavaScripta smatra se 1995. godina kada je kompanija Netscape objavila nekoliko prvih inačica ovog jezika. Izvorni naziv ovog jezika trebao je biti LiveScript, no kako bi se potaknula uporaba novog skriptnog jezika, nazvan je slično programskom jeziku Java, od kojeg se u

tadašnje vrijeme dosta očekivalo. Danas je za standardizaciju skriptnih jezika zadužena organizacija ECMA²². Službeni JavaScript standard je ECMA-262 (Srce, 2015, p. 3). JavaScript nudi korištenje raznih vrsta podataka, funkcija, obrazaca, biblioteka i korisnih skripti. Služi za programiranje HTML stranica, pretvaranje dinamičkog teksta u HTML stranicu, reagiranje na događaje, čitanje i pisanje HTML elemenata, validiranje (provjeru ispravnosti i vjerodostojnosti) podataka, detektiranje preglednika kojeg korisnik upotrebljava i sl. Prednosti JavaScripta su: manja potreba za komunikacijom s poslužiteljem, trenutni odaziv korisniku, povećana interaktivnost i bogatije sučelje. Što se tiče ograničenja, JavaScript je klijentski programski jezik i nije mu dopušteno pisanje ili čitanje datoteka, ne može se koristiti za mrežne aplikacije jer za to ne postoji podrška te nema mogućnost višeprosesorskog izvođenja (Zekić-Sušac, 2009).

Treba znati da JavaScript nije pojednostavljena inačica programskog jezika Java. Jedino što njih povezuje je slična sintaksa i njihovo korištenje za izvršavanje određenih radnji unutar preglednika. JavaScript je samo skriptni jezik, dok je Java pravi objektni programski jezik. Program čija je zadaća obrada i izvršavanje skripte naziva se interpreter ili prevoditelj. Interpreter čita kod i prevodi ga u strojni jezik svakog puta kada se pokrene skripta. Općenito, skriptni se jezici koriste jer je razvoj programa znatno jednostavniji. Za razliku od programa pisanih u pravim programskim jezicima, kod skriptnog jezika ne treba prevoditi skripte u strojni jezik. Dakle, skriptni jezici su programski jezici manjih mogućnosti koji se sastoje od izvršnog računalnog koda, obično ugrađenog u HTML stranice. JavaScript je interpreter, što znači da se skripta izvršava odmah naredbu po naredbu, bez prethodnog prevođenja cijelog programa i kreiranja izvršne datoteke (Chapman, 2017).

Koraci kod programskih jezika:

1. Napisati ili popraviti program
2. Prevesti program u strojni jezik
3. Pokrenuti prevedeni program
4. Za popravke ponoviti od 1. koraka

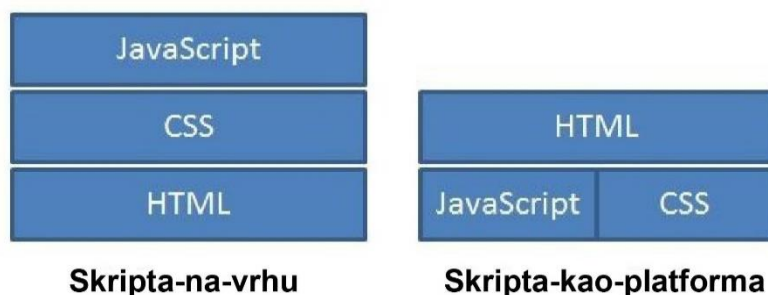
Koraci kod skriptnih jezika:

1. Napisati ili popraviti skriptu

²² engl. European Computer Manufacturers Association

2. Pokrenuti interpreter
3. Za popravke ponoviti od 1. koraka (Srce, 2015, p. 4)

HTML²³ predstavlja prezentacijski jezik koji se koristi za strukturiranje i davanje značenja web sadržaju. HTML se koristi primjerice, kod definiranja odlomaka, naslova, tablica te dodavanja slika i videa na web stranicu. CSS²⁴ je jezik stilskih pravila koji se upotrebljava za stiliziranje HTML sadržaja. Može se koristiti kod postavljanja pozadinskih boja i fontova, postavljanja sadržaja u više stupaca i sl (Mills et al., 2018). Na temelju toga, može se reći da se JavaScript može realizirati kroz dvije arhitekture, a to su Script-on-top i Script-as-a-platform. Script-on-top ili skripta-na-vrhu je slučaj gdje JavaScript kao skripta koristi HTML i CSS te manipulira njima kao komponentama koje čekaju da njima bude upravljano. Script-as-a-platform ili skripta-kao-platforma predstavlja slučaj gdje HTML koristi predefinirano JavaScript ponašanje i CSS stilove (Ince, 2011). Na slici u nastavku su prikazani ti slučajevi.



Slika 26. Skripta-na-vrhu i Skripta-kao-platfoma arhitekture programskog jezika JavaScript
(Izvor: Ince, 2011)

Što se tiče aplikacije e-Tlak, korištenje Java za mobilnu aplikaciju i JavaScripta za web aplikaciju, dobar je odabir kako bi se spoznale temeljne razlike između njih. Dobra stvar je i da na internetu već postoji mnogo gotovih JavaScript skripti koje samo treba znati koristiti i ubaciti u odgovarajući dio. Prema statistici zajednice Stack Overflow, u 2017. godini JavaScript je bio najpopularniji programski jezik koji svoju dominaciju potvrđuje i razvija već nekoliko godina zaredom (Grisogono, 2017).

²³ engl. HyperText Markup Language

²⁴ engl. Cascading Style Sheets

4.2.2. Programski okvir Vue.js

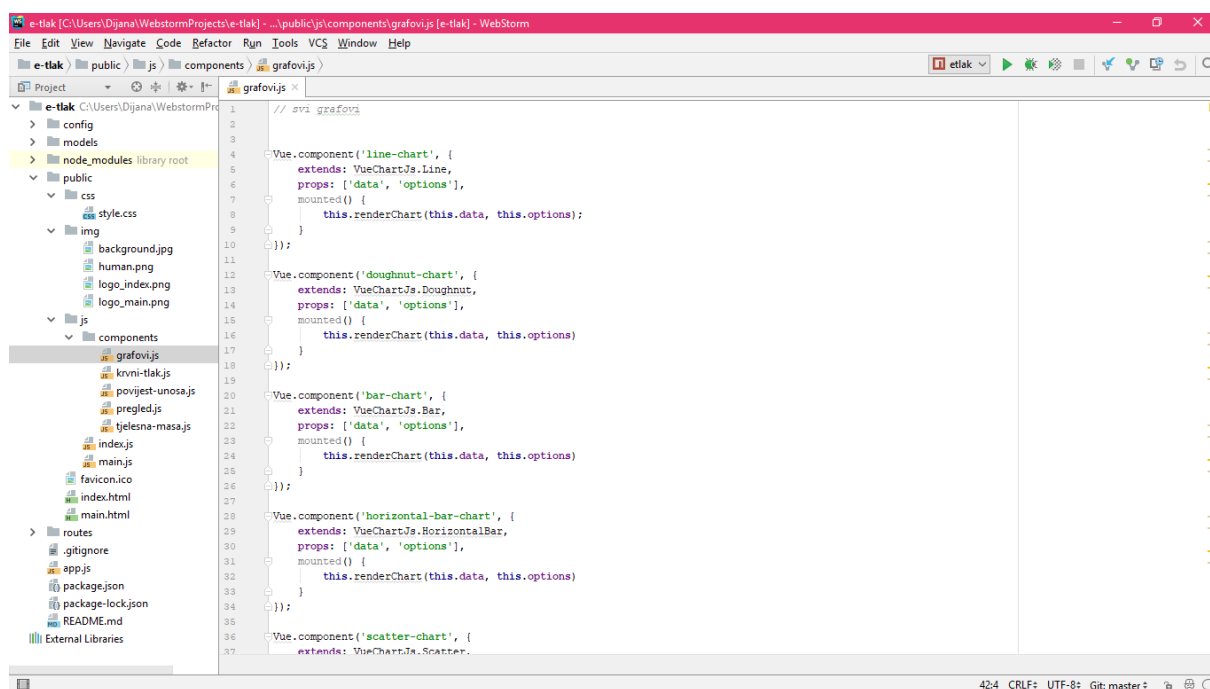
Vue.js predstavlja progresivni JavaScript programski okvir za izradu korisničkog sučelja. Smatra se da je to jedan od najjednostavnijih JavaScript programskih okvira. Vrlo je brz, ne zauzima mnogo i ima detaljno objašnjenu dokumentaciju. Fleksibilan je i prilagodljiv različitim vrstama projekata. Moglo bi se reći da osnovna verzija Vue.js-a predstavlja biblioteku te je tada Vue.js kao takav primjenjiv za male i srednje projekte. Međutim, dodavanjem odgovarajućih komponenti i dodataka, Vue.js može postati pravi potpuni programski okvir te kao takav ima potencijal za izradu poslovnih aplikacija (Chos, 2018). Instalacija Vue.js-a izvršava se kroz Node.js i njegov NPM.

Kod implementacije web aplikacije e-Tlak korišten je već spomenuti IDE WebStorm. Web aplikacija e-Tlak nalazi se u mapi public koju čine dvije podstranice: datoteka index.html koja se odnosi na početnu registraciju i prijavu korisnika te datoteka main.html uz prateće datoteke .js formata koje predstavljaju glavni dio web aplikacije. Vue.js kao glavni programski okvir omogućuje kombinaciju svih naredbi na jednom mjestu. Svaka pojedina komponenta web aplikacije e-Tlak kao i komponente vezane za grafikone, nalaze se u svojoj zasebnoj .js datoteci te se pomoću oznake „<script>“ učitavaju u ruter VueRouter i glavnu Vue instancu prije korištenja. Datoteka main.js sadrži sve definirane rute i komponente te inicijalizirani VueRouter. Točnije, radi se o glavnoj Vue instanci koja sadrži element kojem se pridružuje spomenuti ruter te razne varijable i funkcije. Bavi se učitavanjem redom unesenih imena korisnika, ID-a i tokena korisnika te izvodi i nekoliko funkcija za učitavanje podataka s API-ja koristeći jQuery „\$.ajax“ metodu za asinkrone API pozive. Općenito, jQuery predstavlja biblioteku koja pruža bržu i lakšu implementaciju web aplikacija pisanih u JavaScriptu; pruža mogućnost uz pomoć samo jedne linije koda postići isto što i s desetak ili dvadesetak linija regularnog JavaScript koda. JavaScript uz AJAX²⁵ tehniku omogućuje web stranicama komunikaciju s poslužiteljem (Ivezić, 2010). Slično kao u mobilnoj aplikaciji, šalje se naslov autorizacije s korisnikovim tokenom. Nakon dobivenih podataka s poslužitelja, prevode se datumi iz MongoDB baze podataka u lokalno čitljiv format i to se, zajedno s tekstom naziva dijagnoza iz pomoćne funkcije, sprema u glavnu Vue instancu. Odmah zatim pripremaju se i podaci za tjelesnu masu. Ti su podaci unutar Vue instance i mogu se proslijediti Vue

²⁵ engl. Asynchronous JavaScript and XML

komponentama. Dakle, komponente se odnose na svaku podstranicu web aplikacije e-Tlak: pregled.js, krvni-tlak.js, tjelesna-masa.js i povijest-unosa.js. O funkcioniranju prijave i registracije korisnika na početnoj stranici brine se datoteka index.js.

Za prikaz animiranih i interaktivnih grafikona u sklopu web aplikacije e-Tlak, korištena je biblioteka Chart.js, odnosno VueChartJs kao njegova implementacija za rad u Vue.js-u. Biblioteka omogućuje jednostavno prosljeđivanje podataka i opcija izgleda grafikona u HTML formi (Chart.js, 2018). Komponente vezane za grafikone, prije korištenja su pripremljene u posebnoj datoteci grafovi.js. Nakon toga, grafikoni se koriste pomoću oznaka, npr. „<bar-chart>“ te se detaljnije pripremaju podaci, način strukture podataka i opcije, ovisno o željenom izgledu grafikona. Dakle, za sadržaj web aplikacije e-Tlak zadužen je HTML, dok je za izgled zadužen CSS. Što se tiče CSS-a, osim klasičnog stila navedenog u datoteci style.css, za izgled cijele web aplikacije je najviše zadužen Materialize. On predstavlja poseban jezik namijenjen dizajnu responzivnih web stranica te ima velike sličnosti s dizajnom Googleovih alata kao i modernijih Android aplikacija (Materialize, 2018). Već spomenuti Express.js, hosta web aplikaciju e-Tlak kao običnu web stranicu. Projekt web aplikacije e-Tlak također je dostupan na GitHub platformi, na adresi: <https://github.com/dijana092/e-tlak-web>. U nastavku je prikazan primjer programskog koda u WebStormu koji se odnosi na web aplikaciju e-Tlak, točnije dio s grafikonima.



Slika 27. Primjer programskog koda web aplikacije e-Tlak u IDE-u WebStorm

(Izvor: autorica)

Što se tiče Spring programskog okvira kao solucije, on predstavlja platformu koja pruža bogati skup modula i projekata te široki panel opcija kao podržci razvoju Java web aplikacija. Spring ima svoju primjenu u razvoju velikih aplikacija s kompleksnom poslovnim logikom gdje je nenadmašiv u tom segmentu (Čolak, 2015, p. 35). O kompleksnosti samog Springa govore njegovi važni aspekti, a to su injekcija ovisnosti²⁶ i inverzija kontrole²⁷ kao ključni dio koji omogućuje funkcioniranje Java aplikacije pisane korištenjem Springa. Injekcija ovisnosti predstavlja uzorak dizajna za razvoj softvera koji implementira inverziju kontrole za rješavanje ovisnosti. Ovisan je objekt koji može biti korišten, primjerice servis. Injekcija je prosljeđivanje ovisnosti ovisnom objektu, primjerice klijentu, koji koristi taj objekt. Servis je građevni dio klijenta. Prosljeđivanje servisa klijentu je osnovni zahtjev uzorka dizajna (Prasanna, 2009). Inverzija kontrole u softverskom inženjerstvu opisuje dizajn u kojem posebno pisani dijelovi programa primaju tok kontrola od generičkih biblioteka. Arhitektura softvera s ovim dizajnom je obrnula kontrolu za razliku od tradicionalnog proceduralnog programiranja. U tradicionalnom programiranju, posebno pisani kod koji izražava svrhu programa poziva biblioteku da se pobrine za generičke zadatke,

²⁶ engl. Dependency injection

²⁷ engl. Inversion of Control

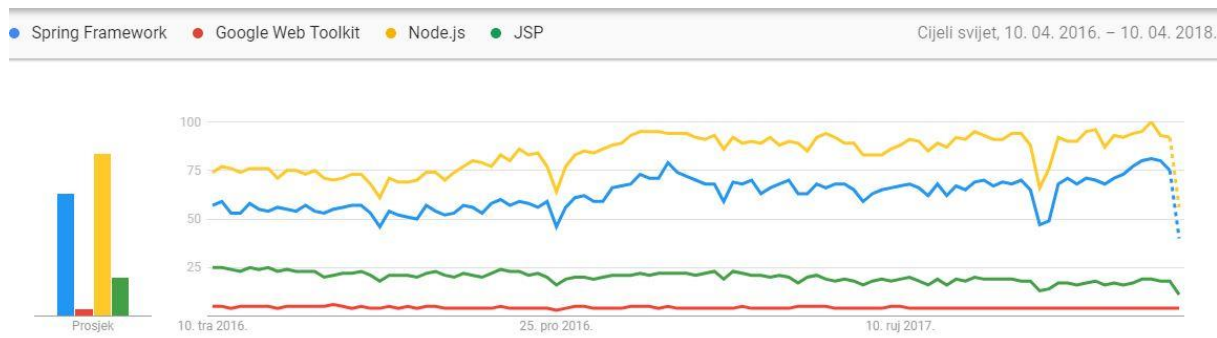
dok s inverzijom kontrole, kod koji se iznova koristi poziva se u posebno pisani ili kod specifičan za zadatak. Inverzija kontrole se koristi da poveća modularnost programa i učini ga proširivim. Inverzija kontrole je povezana s principom injekcije ovisnosti, ali je različita od njega po pitanju razdvajanja ovisnosti između slojeva visoke razine i slojeva niske razine kroz dijeljene apstrakcije. Nedostatak je što injekcija ovisnosti može otežati čitanje koda jer razdvaja ponašanje od izgradnje. To znači da programer mora posvetiti više vremena dokumentaciji kako bi proučio način kako sustav funkcionira. Osim toga, injekcija ovisnosti zahtijeva više planiranja načina na koji će se pisati kod i implementirati jer klijent ne može pozivati direktno nego mora dati zahtjev servisu i servis mora odobriti i osigurati da je klijent dobio potrebne dijelove (Čolak, 2015, pp. 12-15).

Iako pruža bogatstvo funkcionalnosti za razvoj aplikacija, Spring je manjkav u smislu organizacije osnovnih građevnih blokova u jedinstvenu i funkcionalnu cjelinu i taj zadatak pada na programera. Također, nedostatak je i mali broj uputa i dokumentacije te je Spring vezan isključivo za Java programski jezik (Ibid, pp. 2-5). Ako se radi o programeru početniku, prilično je težak za učenje zato što sadrži čitav niz programskih metoda i detalja koji zahtijevaju velik utrošak vremena za razumijevanje svakog pojedinog dijela kao i razumijevanje XML-a jer se ovdje prvenstveno radi o kodiranju u XML obliku. Spring sadrži preko 2400 klasa, oko 49 raznih alata te nema jasnog fokusa što ga može učiniti dosta kompliciranim (Aswani, 2014). Veliki broj modula može biti i nedostatak u slučaju da se rade manje kompleksne aplikacije. Definitivno nije namijenjen za male i srednje aplikacije koje se ne planiraju dalje pretjerano razvijati i unaprjeđivati jer u tom slučaju postoje puno bolje tehnologije koje će omogućiti puno brže obavljanje posla. Spring je najbolja moguća tehnologija za razvoj aplikacije koja će biti održavana svakodnevno (Čolak, 2015, p. 35). Što se tiče e-Tlak aplikacije, mobilna aplikacija obuhvaća Javu, a web aplikacija se odnosi na JavaScript te ostale suvremene tehnologije što stvara raznolikost u učenju i znanju, dok bi u slučaju Springa fokus bio samo na Javu i u mobilnom i u web dijelu aplikacije. Dakle, Spring je jedna od najpopularnijih tehnologija namijenjena za razvoj kompleksnih poslovnih aplikacija, cijelih portala, za povezivanje aplikacije na neku društvenu mrežu i sl., a e-Tlak ne spada u tu skupinu.

Tomcat predstavlja aplikacijski poslužitelj u vlasništvu kompanije Apache Software Foundation temeljen na programskom jeziku Java, a namijenjen prikazu web stranica

koje uključuju kodiranje Java poslužitelja. U slučaju Tomcata, često je komplicirano izmijeniti konfiguracije poslužitelja kao i koristiti ažuriranja što može produžiti vrijeme samog razvoja aplikacije (TrustRadius, 2016), dok je s prethodno opisanim odabirom tehnologije vezanim za aplikaciju e-Tlak to puno lakše i intuitivnije. Node.js, odnosno Vue.js kao programski okvir svojim specifičnim načinom rada izbjegao je potrebu za stvaranjem posebnog novog projekta kao što bi se to moralo uz neke druge programske okvire. Uz VueRouter se smanjilo ponavljanje koda i API poziva kao i potreba za nekim određenim komandno-linijskim alatom. Inače, obične poveznice bi vodile na druge .html datoteke, s ovako se to obavlja brzo i bez ponovnog učitavanja web aplikacije. Ovakav odabir tehnologije je bolji također i iz razloga što Express.js odrađuje ulogu vezanu i za API i za web poslužitelj.

Zanimljiv programski okvir za razvoj, izradu i optimizaciju složenijih aplikacija namijenjenih internetskim preglednicima je i Google Web Toolkit. On omogućuje izradu web aplikacija visokih performansi u Java programskom jeziku programerima koji nisu toliko upoznati s tehnologijama. Navedena tehnologija ubrzava razvoj web aplikacije, smanjuje količinu tehnologija koju je potrebno koristiti za izradu završnog proizvoda te olakšava kasnije održavanje i nadogradnju (Golub et al., 2013). Glavna prednost je što programer s iskustvom u Javi može vrlo brzo naučiti lako razvijati aplikacije kroz Google Web Toolkit. Nedostatak je što postoji mnogo verzija Google Web Tookita, pa je često potrebno korigirati njegove funkcije, sučelja i događaje. Također, postoji samo Google dokumentacija, konkretnih knjiga gotovo da i nema. Sve je u Javi što nije pretjerano „zabavno“ za onog tko želi naučiti još ponešto dodatno od jezika i tehnologija (Marwaha, 2012). Google Web Toolkit je puno manje popularan od Springa i ne nudi mnogo novog i raznolikog znanja. Stoga su Node.js i Vue.js „zlatna sredina“ između Springa i Google Web Tookita, a za potrebe e-Tlak aplikacije. Programskih okvira je zaista mnogo. Još jedan od njih je i Java Server Pages koji predstavlja Java programski okvir koji omogućuje izradu cijelih portala. No, zastario je, a zbog novijih programskih okvira sve više gubi na popularnosti. U nastavku je prikazan grafikon interesa za Spring, Google Web Toolkit, Node.js i Java Server Pages u razdoblju od 2016.-2018. godine od strane Google Trends istraživanja.



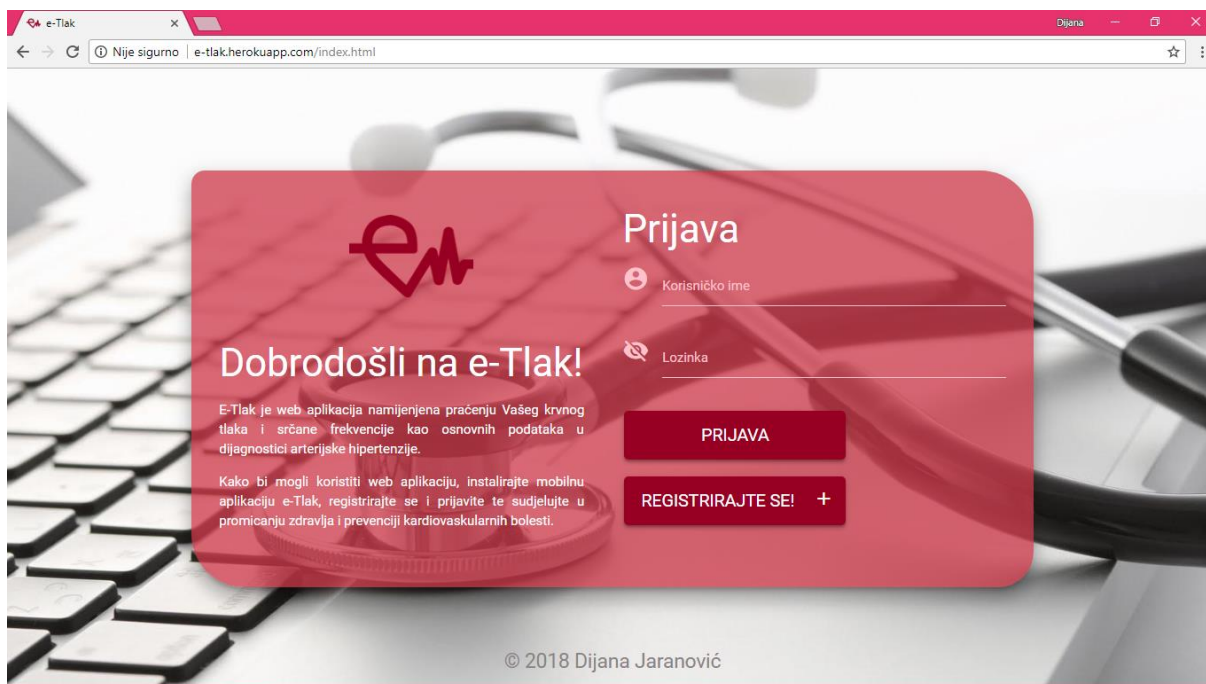
Slika 28. Interes za programskim okvirima Spring, Google Web Toolkit, Node.js i Java Server Pages (Izvor: Google Trends, 2018)

4.3. Korisničke upute

Web aplikacija e-Tlak ističe se raznim vizualizacijama, grafikonima, detaljima te jednostavnom navigacijom. Dizajn je kao i u mobilnoj aplikaciji, radi se o mekim i pastelnim nijansama crvene boje i bijele boje. Dostupna je na web adresi: <http://e-tlak.herokuapp.com>.

4.3.1. Prijava

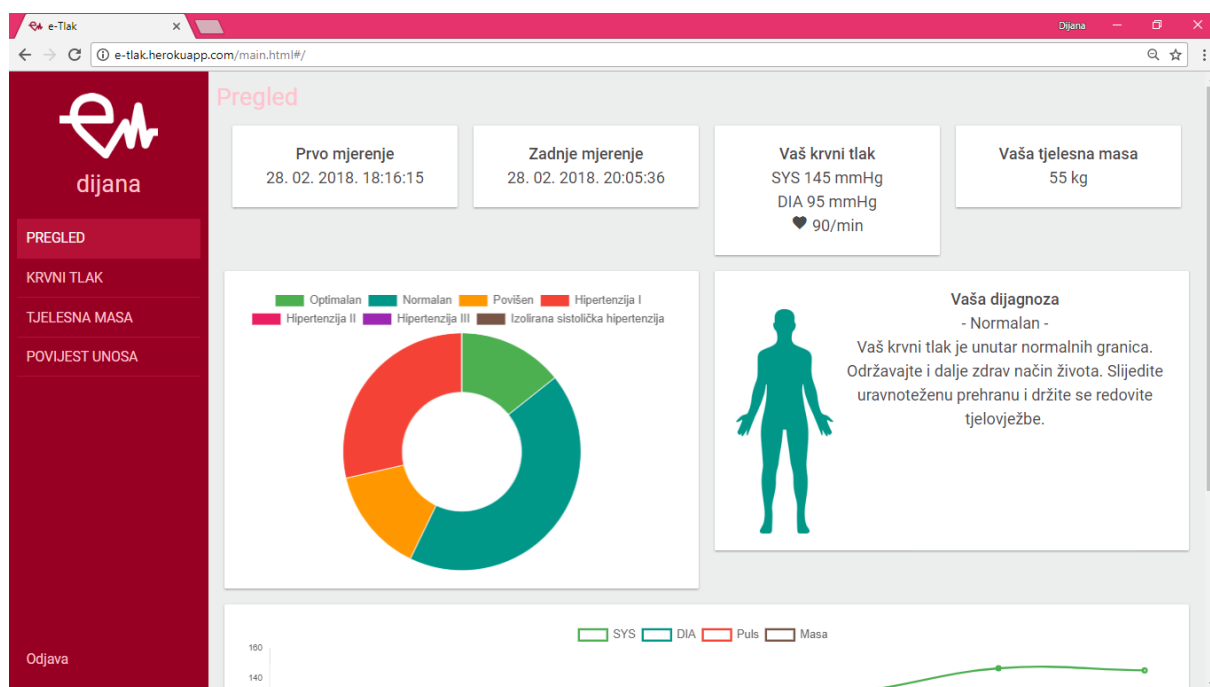
Prvi zaslon web aplikacije e-Tlak sadrži kraći opis aplikacije i formu za unos korisničkog imena i lozinke vezano za prijavu. Unosi se korisničko ime i lozinka koji su se koristili prilikom prijave u mobilnu aplikaciju e-Tlak. U slučaju da nije bilo registracije kroz mobilnu aplikaciju, može se to učiniti i putem web aplikacije, klikom na gumb „Registrirajte se!“



Slika 29. Početni zaslon web aplikacije e-Tlak: Prijava
(Izvor: autorica)

4.3.2. Pregled dijagnoze

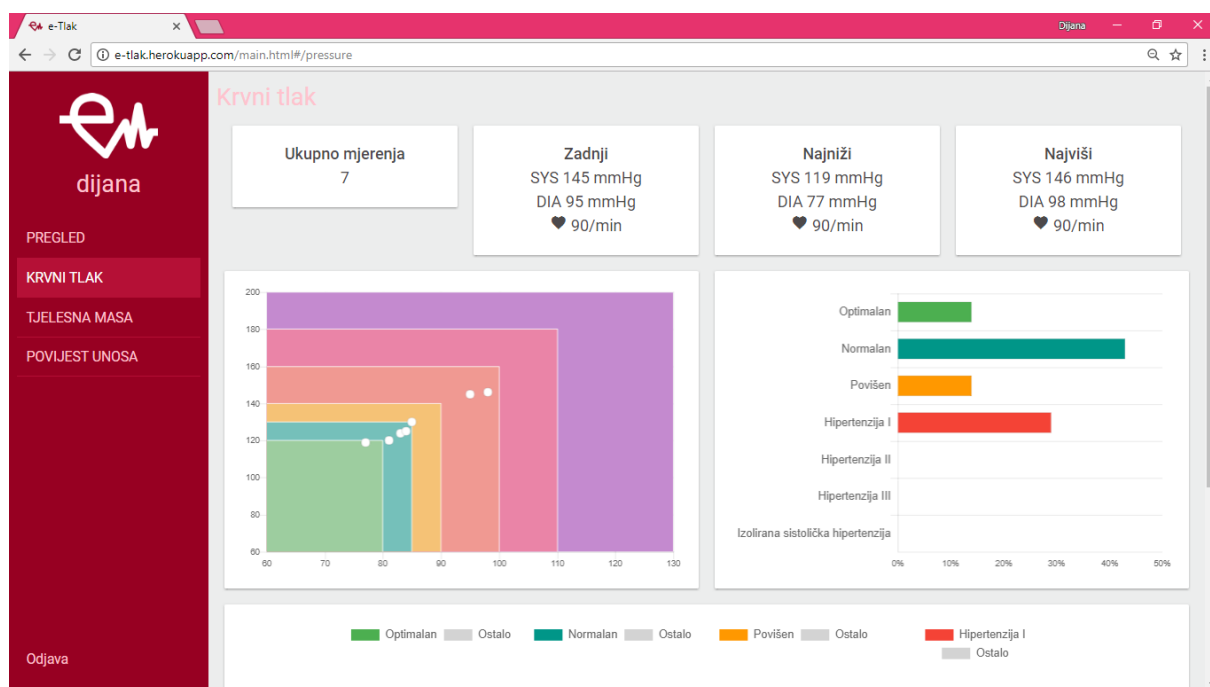
Prijavom u web aplikaciju e-Tlak dolazi se na glavnu stranicu koja nudi četiri podstranice od kojih je prva Pregled. Ona sadrži sažetak dijagnoze gdje se može vidjeti vrijeme prvog i zadnjeg mjerenja krvnog tlaka, podatak o trenutnom krvnom tlaku i trenutnoj tjelesnoj masi. Vidljiv je i tortni grafikon dijagnoze koji prikazuje točan postotak svake prisutne dijagnoze te vizualno predočava najprisutniju dijagnozu. Desno od grafikona, nalazi se skica ljudske siluete koja mijenja boju ovisno o dijagnozi te sadrži opis dijagnoze i savjeta kojih bi se trebalo pridržavati. Ispod se nalazi linijski grafikon čiji je fokus na sistolički krvni tlak, dijastolički krvni tlak, puls i tjelesnu masu, odnosno vrijednost, točan datum i vrijeme unesenog. Klikom na pojedini element legende grafikona, moguće je ukloniti prikaz željenog podatka.



Slika 30. Glavni zaslon web aplikacije e-Tlak: kartica Pregled
(Izvor: autorica)

4.3.3. Pregled krvnog tlaka

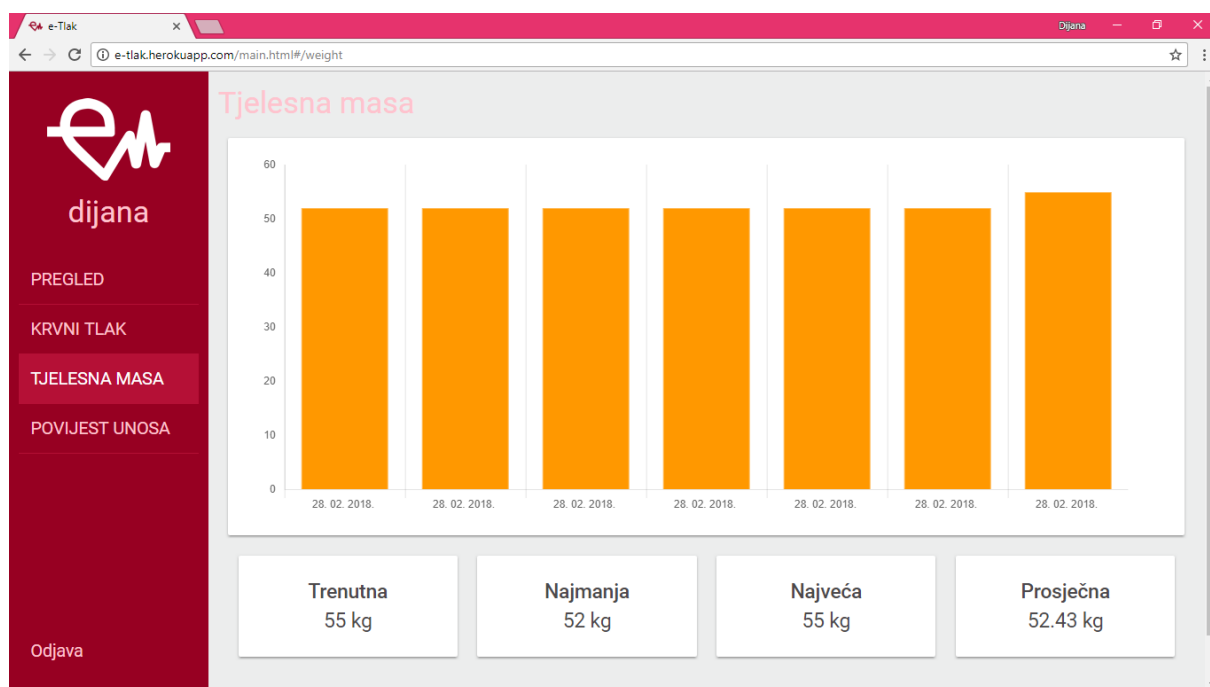
Druga kartica Krvni tlak sadrži detalje o krvnom tlaku: ukupno mjerenja, vrijednosti zadnjeg unesenog krvnog tlaka te najnižeg i najvišeg krvnog tlaka. Raspršeni grafikon sadrži sve unose i pozadinu koja predstavlja granice dijagnoza, kako bi se mogle vidjeti točne vrijednosti sistoličkog i dijastoličkog krvnog tlaka i iščitati kojoj točno dijagnozi unos pripada te da li je na granici. Desno se nalaze točni postotci svake pojedine dijagnoze. Ispod su tortni grafikoni za svaku prisutnu dijagnozu, tj. odnos dijagnoze naspram svih ostalih dijagnoza.



Slika 31. Glavni zaslon web aplikacije e-Tlak: kartica Krvni tlak
(Izvor: autorica)

4.3.4. Pregled tjelesne mase

Kartica vezana za pregled tjelesne mase nudi grafički prikaz unesene tjelesne mase kroz vrijeme, eventualne promjene i oscilacije. Na dnu su ispisane vrijednosti trenutne, najmanje, najveće i prosječne tjelesne mase.



Slika 32. Glavni zaslon web aplikacije e-Tlak: kartica Tjelesna masa
(Izvor: autorica)

4.3.5. Pregled povijesti unosa

Povijest unosa je posljednja kartica web aplikacije e-Tlak koja omogućuje filtriranje i prikaz svih unosa. Kod filtriranja se odabire početni datum od kojeg se žele vidjeti unosi, kao i posljednji. Nakon toga, izlistava se tablični prikaz s rednim brojem unosa, točnim datumom i vremenom, vrijednostima sistoličkog tlaka, dijastoličkog tlaka i pulsa, određenom dijagnozom i vrijednosti tjelesne mase. Moguće je označiti kvačicama onaj element ili više elemenata čiji se prikaz želi vidjeti u tablici. Također, moguće je selektiranjem određenog naslova stupca tablice posložiti prikaz po željenom redoslijedu.

Povijest unosa

Filtriraj: Od: 01.01.2018. Do: 31.12.2018.

Prikaži: ☒ Sistolički ☒ Dijastolički ☒ Puls ☒ Dijagnoza ☒ Tjelesna masa

R.br.	Datum	Vrijeme	Sistolički	Dijastolički	Puls	Dijagnoza	Masa
1	28. 02. 2018.	18:16:15	119	77	80	optimalan	52
2	28. 02. 2018.	18:20:00	120	81	93	normalan	52
3	28. 02. 2018.	18:25:09	124	83	91	normalan	52
4	28. 02. 2018.	18:29:59	125	84	89	normalan	52
5	28. 02. 2018.	18:35:01	130	85	95	povisen	52
6	28. 02. 2018.	18:40:01	146	98	90	hipertenzija I	52
7	28. 02. 2018.	20:05:36	145	95	90	hipertenzija I	55

Slika 33. Glavni zaslon web aplikacije e-Tlak: kartica Povijest unosa
(Izvor: autorica)

ZAKLJUČAK

Ovaj diplomski rad obuhvaća jedan primjer razvoja aplikacije za praćenje krvnog tlaka. Uz teorijski i praktični dio razvoja aplikacije, stvorene su mobilna aplikacija e-Tlak i web aplikacija e-Tlak kao aplikacije prilagođene korisnicima namijenjene onima koji boluju od arterijske hipertenzije ili sudjeluju u njenoj prevenciji. Kvalitetno vođenje svih unosa krvnog tlaka uz bogate vizualizacije dijagnoze, stvara kod korisnika motivaciju za brigu o zdravlju.

Za ovakve tzv. „user-friendly“ aplikacije prilagođene korisnicima potrebna je i „user-friendly“ tehnologija. Danas je objektno orijentirano programiranje jedna od najvažnijih programskih paradigmi. Programer je oslobođen razmišljanja o implementacijskim detaljima te mu se uz tzv. programske okvire nude gotove funkcionalnosti kako bi se omogućila ponovna iskoristivost koda. Ideja je da se programira jednom, a koristi višestruko te je fokus na rješavanje problema koji se vezuju uz namjenu same aplikacije. Provedenom analizom i istraživanjem uvidio se najefikasniji način realizacije i odabir tehnologije za izradu aplikacije e-Tlak. Iz tog razloga je odabrana Java kao primjer čistog objektnog jezika, a isto tako i JavaScript kao popularan jezik za front-end i back-end dio aplikacije koji nudi izradu bogatijeg sučelja i koji je primjenjiv u raznim razvojnim okolinama. Jedan od glavnih pokretača ekspanzije popularnosti JavaScripta je Googleov brzi V8 sustav uz pomoć kojeg se danas može vidjeti izuzetno brzo pokretanje JavaScripta u preglednicima (Chrome, Opera), na poslužitelju (Node.js), u bazama podataka te u brojnim aplikacijama. Nekad je za početak dovoljno samo znanje HTML-a i CSS-a te sve što tada treba pronaći je nešto za spajanje, odnosno ponašanje i logiku, a za to je Vue.js idealan izbor.

Node.js se može nazvati „user-friendly“ alatom jer posjeduje sve karakteristike za to. Programski kod napisan na Node.js platformi ne slijedi tradicionalni model primanja, obrade, slanja i čekanja podataka. Može se reći kako će Node.js sigurno preuzeti dio tržišta jer se razvija velikom brzinom te postaje sve više konkurentan. Vrlo je bitno koristiti alate koji pojednostavljuju sam tijek i proces programiranja. „User-friendly“ alati uvelike pomažu i štede korisniku vrijeme te se on može posvetiti samoj logici programa. Aplikacije poput e-Tlak mobilne i web aplikacije implementiranih uz pomoć „user-friendly“ alata će zasigurno prevladati kao glavni korisnikovi izbori svakodnevice, bilo to na mobilnom uređaju ili internetskom pregledniku.

LITERATURA

1. ADARSH (2015) *What is difference between JDK, JRE and JVM?*, Quora, Dostupno na: <https://www.quora.com/What-is-difference-between-JDK-JRE-and-JVM> [Pristupljeno: 4.2.2018.]
2. ANDROID DEVELOPERS (2018) *Android, the world's most popular mobile platform*, Dostupno na: <https://developer.android.com/about/index.html> [Pristupljeno: 6.2.2018.]
3. ANDROID DEVELOPERS (2018) *Platform Architecture*, Dostupno na: <https://developer.android.com/guide/platform/index.html> [Pristupljeno: 16.2.2018.]
4. ANDROID DEVELOPERS (2018) *Transmitting Network Data Using Volley: Sending a Simple Request*, Dostupno na: <https://developer.android.com/training/volley/simple.html> [Pristupljeno: 17.2.2018.]
5. ASWANI (2014) *Spring Framework Advantages and Disadvantages*, One Stop Blog, Dostupno na: <http://www.aksindiblog.com/spring-framework-advantages-disadvantages.html> [Pristupljeno: 6.4.2018.]
6. BCRYPT (2018) *Bcrypt.js*, Dostupno na: <https://www.npmjs.com/package/bcryptjs> [Pristupljeno: 27.2.2018.]
7. BRDAR, L. (2016) *REST API kao poslužiteljski dio jednostranične web aplikacije*, Završni rad, Sveučilište u Rijeci, Tehnički fakultet, Dostupno na: https://bib.irb.hr/datoteka/835028.0069065248_1035_Brdar_Leo.pdf [Pristupljeno: 3.2.2018.]
8. CAMPBELL, O. (2015) *Designing For The Elderly: Ways Older People Use Digital Technology Differently*, Smashing Magazine, Dostupno na: <https://www.smashingmagazine.com/2015/02/designing-digital-technology-for-the-elderly/> [Pristupljeno: 29.1.2018.]
9. CHANDRAYAN, P. (2017) *All About Node.js You Wanted To Know?*, CodeBurst, Dostupno na: <https://codeburst.io/all-about-node-js-you-wanted-to-know-25f3374e0be7> [Pristupljeno: 27.2.2018.]

10. CHAPMAN, S (2017) *Introduction to JavaScript*, Computer Science, ThoughtCo, Dostupno na: <https://www.thoughtco.com/what-is-javascript-2037921> [Pristupljeno: 18.3.2018.]
11. CHART.JS (2018) *Simple yet flexible JavaScript charting for designers & developers*, Dostupno na: <http://www.chartjs.org/> [Pristupljeno: 21.2.2018.]
12. CHOS (2018) *Životni ciklus Vue.js instance/komponente*, Web Programiranje, Dostupno na: <https://www.webprogramiranje.org/tag/vue-js/> [Pristupljeno: 22.3.2018.]
13. CORS (2018) *Cross-Origin Resource Sharing (CORS)*, Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS> [Pristupljeno: 27.2.2018.]
14. ČOLAK, S. (2015) *Izrada web aplikacije u razvojnom okruženju Spring*, Završni rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin, Dostupno na: https://bib.irb.hr/datoteka/717071.1-Izrada_web_aplikacije_u_razvojnom_okruenju_Spring.pdf [Pristupljeno: 25.2.2018.]
15. DI MARZIO, J. F. (2017) *Android Programming with Android Studio*, Indianapolis: John Wiley & Sons
16. ERCEG, M. et al. (2013) *Javnozdravstveni značaj arterijske hipertenzije*, Hrvatski časopis za javno zdravstvo, vol. 9, broj 34, Zagreb: Hrvatski zavod za javno zdravstvo, Dostupno na: <http://hcjz.hr/index.php/hcjz/article/view/195/177> [Pristupljeno: 11.1.2018.]
17. EXPRESS (2018) *Fast, unopinionated, minimalist web framework for Node.js*, Dostupno na: <https://expressjs.com/> [Pristupljeno: 27.2.2018.]
18. GAČIĆ, J. (2017) *NoSQL baze podataka*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Dostupno na: <https://zir.nsk.hr/islandora/object/pmf%3A3234/datastream/PDF/view> [Pristupljeno: 27.2.2018.]
19. GARRETT, J. (2011) *The Elements of User Experience*, SAD: New Riders
20. GOLUB, B. et al. (2013) *Programska podrška za razmjenu i analizu ostvarenih elektroenergetskih veličina*, Cigre, Dostupno na: <http://www.koncar->

- ket.hr/docs/koncarketHR/documents/140/1_0/Original.pdf [Pristupljeno: 5.4.2018.]
21. GOOGLE PLAY (2018) Dostupno na:
<https://play.google.com/store/apps/details?id=com.medm.medmbp.diary>
[Pristupljeno: 28.1.2018.]
22. GOOGLE TRENDS (2018) Dostupno na:
<https://trends.google.com/trends/explore> [Pristupljeno: 6.4.2018.]
23. GRISOGONO, A. (2017) *Kako je i zašto JavaScript postao ikona među programskim jezicima*, Netokracija, Dostupno na:
<http://www.netokracija.com/dump-days-2017-javascript-135899> [Pristupljeno: 21.3.2018.]
24. HEROKU (2018) *Heroku Postgres PGX: New Plans, More Possibilities*, Dostupno na: <https://www.heroku.com/> [Pristupljeno: 27.2.2018.]
25. HORVAT, D. (2016) *Dijagnostika–Kako interpretirati nalaz kontinuiranog mjerenja arterijskog tlaka?*, Pliva, Dostupno na:
<http://www.plivamed.net/medicus/clanak/12286/DijagnostikaKako-interpretirati-nalaz-kontinuiranog-mjerenja-arterijskog-tlaka.html> [Pristupljeno: 20.1.2018.]
26. INCE, S. J. (2011) *JavaScript Architecture*, Microsoft Developer, Dostupno na:
<https://blogs.msdn.microsoft.com/simonince/2011/02/28/javascript-architecture/> [Pristupljeno: 20.3.2018.]
27. IVEZIĆ, J. (2010) *JQuery - Što je i čemu služi?*, PopArt Studio, Dostupno na:
https://www.popwebdesign.net/popart_blog/2010/06/jquery-sta-je-i-cemu-sluzi/
[Pristupljeno: 23.3.2018.]
28. JANJANIN, B. (2011.) *Nemojte ponoviti 5 najčešćih grešaka u izradi sučelja vaše Android, Symbian ili iPhone aplikacije*, Netokracija, Dostupno na:
<http://www.netokracija.com/najcesce-greske-izrada-sucelja-android-symbian-iphone-aplikacija-9245> [Pristupljeno: 29.1.2018.]
29. JARANOVIĆ, D. (2015) *Mobilne aplikacije i usluge računalstva u oblaku*, Završni rad, Sveučilište Jurja Dobrile u Puli, Fakultet ekonomije i turizma "Dr. Mijo Mirković", Dostupno na: <https://zir.nsk.hr/islandora/object/unipu:188>
[Pristupljeno: 29.1.2018.]

30. JSON WEB TOKENS (2018) *Introduction to JSON Web Tokens*, Dostupno na: <https://jwt.io/> [Pristupljeno: 27.2.2018.]
31. KRALJ, V. (2016) *Svjetski dan hipertenzije 2016.*, HZJZ, Dostupno na: <http://javno-zdravlje.hr/event/svjetski-dan-hipertenzije-2016/> [Pristupljeno: 7.1.2018.]
32. KVANTUM-TIM (2014) *Samokontrola visokog krvnog tlaka učinkovita je kod visokorizičnih pacijenata*, Dostupno na: <http://lijecnici.com.hr/197-samokontrola-visokog-krvnog-tlaka-ucinkovita-je-kod-visokorizicnih-pacijenata/> [Pristupljeno: 19.1.2018.]
33. LEE, S. et al. (2009.) *The Performance of Touch Screen Soft Buttons. Clicking on Buttons*. SAD: CHI, Dostupno na: http://www.researchgate.net/profile/Seungyon_Lee/publication/221514597_The_performance_of_touch_screen_soft_buttons/links/5473c3580cf245eb436db968.pdf [Pristupljeno: 29.1.2018.]
34. LOUREIRO, B. et al. (2014) *Design Guidelines and Design Recommendations of Multi-Touch Interfaces for Elders*, The Seventh International Conference on Advances in Computer-Human Interactions
35. LJUBIĆ, S. (2011.) *Upotrebljivost mobilnih aplikacija*, Sveučilište u Rijeci, Tehnički fakultet, Dostupno na: https://www.fer.hr/download/repository/KDI-Ljubic-Upotrebljivost_mobilnih_aplikacija.pdf [Pristupljeno: 20.2.2018.]
36. MARWAHA, G. (2012) *GWT – Pros and Cons*, Java Code Geeks, Dostupno na: <https://www.javacodegeeks.com/2012/01/gwt-pros-and-cons.html> [Pristupljeno: 5.4.2018.]
37. MATERIALIZE (2018) *Learn how to easily start using Materialize in your website*, Dostupno na: <http://materializecss.com/about.html> [Pristupljeno: 23.2.2018.]
38. MEDIKOR (2016) *Aplikacija Medikor provjera zdravlja: moderan pristup kontroli zdravstvenog stanja*, Dostupno na: <http://medikor.hr/wp/aplikacija-medikor-provjera-zdravlja-moderan-pristup-kontroli-zdravstvenog-stanja/> [Pristupljeno: 19.1.2018.]

39. MEŠTROVIĆ, A. (2015) *Izazov za ljekarnika - kardiovaskularni pacijenti*, inPharma, Dostupno na:
<http://www.inpharma.hr/index.php/news/94/20/Izazov-za-ljekarnika-kardiovaskularni-pacijenti> [Pristupljeno: 17.1.2018.]
40. MILLS, et al (2018) *What is JavaScript?*, MSDN Web Docs, Dostupno na:
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript [Pristupljeno: 21.3.2018.]
41. MLAB (2018) *Trusted. Loved. Most widely deployed*, Dostupno na:
<https://mlab.com/> [Pristupljeno: 27.2.2018.]
42. MONGOOSE (2018) *Elegant mongodb object modeling for node.js*, Dostupno na: <http://mongoosejs.com/> [Pristupljeno: 27.2.2018.]
43. MOORE WILLIAMS, J. (2017) *10 essential UI (user-interface) design tips*, Webflow Blog, Dostupno na: <https://webflow.com/blog/10-essential-ui-design-tips> [Pristupljeno: 27.1.2018.]
44. MSD PRIRUČNIK. (2014) *Arterijska hipertenzija*, Dostupno na:
<http://www.msd-prirucnici.placebo.hr/msd-prirucnik/kardiologija/arterijska-hipertenzija> [Pristupljeno: 5.1.2018.]
45. PAL, P. (2016) *How To Design Dashboards More User-Friendly?*, Think 360, Dostupno na: <https://think360studio.com/how-to-design-dashboard-more-user-friendly/> [Pristupljeno: 29.1.2018.]
46. PENEZIĆ, D. (2013) *Node.js - platforma za jednostavnu izradu skalabilnih mrežnih aplikacija*, SISTEMAC, Dostupno na:
http://www.srce.unizg.hr/arhiva_weba/sistematic2015/index.php%3Fid=35&no_cache=1&tx_ttnews%255Btt_news%255D=976.html [Pristupljeno: 15.2.2018.]
47. PHILLIPS, B. et al. (2015) *Android Programming: The Big Nerd Ranch Guide*, Atlanta: Big Nerd Ranch
48. PLAZONIĆ, T. (2016) *Web aplikacija za upravljanje službenim putovanjima*, Diplomski rad, Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički fakultet, Dostupno na:

<https://repozitorij.etfos.hr/islandora/object/etfos%3A780/datastream/PDF/view>

[Pristupljeno: 27.2.2018.]

49. POPIĆ, F. (2015) *Aplikacija za brzo dogovaranje sastanaka pametnim telefonima s operacijskim sustavom Android*, Završni rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Dostupno na: https://bib.irb.hr/datoteka/768455.Final_0036470209_39.pdf [Pristupljeno: 18.2.2018.]
50. PRASANNA, D. (2009) *Dependency injection*, Greenwich, Conn.: Manning
51. RAGUŽ, R. (2017) *Prednosti i nedostaci Node.js platforme*, Završni rad, Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i telekomunikacija, Dostupno na: <https://zir.nsk.hr/islandora/object/etfos%3A1267/datastream/PDF/view> [Pristupljeno: 27.2.2018.]
52. SAVITCH, W. (2016) *Absolute Java*, Engleska: Pearson Education Limited
53. SCIENCE DAILY. (2017) *Do you really have high blood pressure?*, Dostupno na: <https://www.sciencedaily.com/releases/2017/03/170320085054.htm> [Pristupljeno: 15.1.2018.]
54. SLAVIČEK, T. (2014) *Designing a mobile interface for older people*, Medium, Dostupno na: <https://medium.com/@tomasslavicek/designing-a-mobile-interface-for-older-people-1c9b70fd645c> [Pristupljeno: 27.1.2018.]
55. SRCE (2015) *Osnove JavaScripta*, Sveučilište u Zagrebu, Sveučilišni računarski centar, Dostupno na: http://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf [Pristupljeno: 10.3.2018.]
56. STATISTICS TIMES (2017) *Top Computer Languages*, Dostupno na: <http://statisticstimes.com/tech/top-computer-languages.php> [Pristupljeno: 14.2.2018.]
57. TOŠIĆ, K. (2015) *Što je Node.js?* PopArt Studio, Dostupno na: https://www.popwebdesign.net/popart_blog/2015/06/sta-je-node-js/ [Pristupljeno: 27.2.2018.]

58. TRUST RADIUS (2016) *User Review: "Tomcat Servers"*, Dostupno na: <https://www.trustradius.com/products/apache-tomcat/reviews/pros-and-cons?cg=small> [Pristupljeno: 26.2.2018.]
59. TUTORIALS POINT (2014) *Java Tutorial*, Tutorials Point
60. WORLD HEALTH ORGANIZATION. (2013) *Why hypertension is a major public health issue*, Švicarska: A global brief on hypertension, Dostupno na: http://ish-world.com/downloads/pdf/global_brief_hypertension.pdf [Pristupljeno: 15.1.2018.]
61. ZEKIĆ-SUŠAC, M (2009) *JavaScript: Uvod, varijable, naredbe, petlje*, Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku, Dostupno na: http://www.mathos.unios.hr/wp/wp2009-10/P8_Java.pdf [Pristupljeno: 12.3.2018.]

POPIS SLIKA

Slika 1. Glavni zaslone mobilne aplikacije MedM Blood Pressure	7
Slika 2. Glavni zaslon web aplikacije Medikor	8
Slika 3. Klasni dijagram sustava e-Tlak	17
Slika 4. Dijagram komponenata sustava e-Tlak.....	18
Slika 5. Arhitektura platforme Node.js.....	20
Slika 6. Primjer progr. koda back-enda aplikacije e-Tlak u IDE-u WebStorm	22
Slika 7. Primjer kolekcije baze podataka aplikacije e-Tlak u mLab bazi podataka....	24
Slika 8. Dijagram slučajeva korištenja mobilne aplikacije e-Tlak	26
Slika 9. Sekvencijski dijagram mobilne aplikacije e-Tlak	28
Slika 10. Zaslone mobilne aplikacije Dnevnik krvnog tlaka	30
Slika 11. Arhitektura programskog jezika Java	32
Slika 12. Arhitektura operacijskog sustava Android	34
Slika 13. Primjer progr. koda mobilne aplikacije e-Tlak u IDE-u Android Studio.....	36
Slika 14. Zaslon s logotipom i zaslone registracije i prijave mob. aplikacije e-Tlak....	37
Slika 15. Glavni zaslon i zaslone za unos tlaka i mase mobilne aplikacije e-Tlak.....	38
Slika 16. Zaslone prikaza dijagnoze mobilne aplikacije e-Tlak	39
Slika 17. Zaslone pregleda povijesti unosa mobilne aplikacije e-Tlak.....	39
Slika 18. Dijagram slučajeva korištenja web aplikacije e-Tlak	41
Slika 19. Sekvencijski dijagram web aplikacije e-Tlak	42
Slika 20. Prototip sučelja web aplikacije e-Tlak: Registracija korisnika.....	43
Slika 21. Prototip sučelja web aplikacije e-Tlak: Prijava korisnika	44
Slika 22. Prototip sučelja web aplikacije e-Tlak: kartica Pregled	44
Slika 23. Prototip sučelja web aplikacije e-Tlak: kartica Krvni tlak	45
Slika 24. Prototip sučelja web aplikacije e-Tlak: kartica Tjelesna masa.....	45
Slika 25. Prototip sučelja web aplikacije e-Tlak: kartica Povijest unosa.....	46

Slika 26. Skripta-na-vrhu i Skripta-kao-platfoma arhitekture progr.jezika JavaScript	48
Slika 27. Primjer programskog koda web aplikacije e-Tlak u IDE-u WebStorm.....	51
Slika 28. Interes za programskim okvirima Spring, GWT, Node.js i JSP	54
Slika 29. Početni zaslon web aplikacije e-Tlak: Prijava	55
Slika 30. Glavni zaslon web aplikacije e-Tlak: kartica Pregled	56
Slika 31. Glavni zaslon web aplikacije e-Tlak: kartica Krvni tlak	57
Slika 32. Glavni zaslon web aplikacije e-Tlak: kartica Tjelesna masa.....	58
Slika 33. Glavni zaslon web aplikacije e-Tlak: kartica Povijest unosa.....	59

POPIS TABLICA

Tablica 1. Kategorije arterijskog tlaka	5
---	---

PRILOZI

CD s digitalnom verzijom rada i implementacijama:

- mobilna aplikacija e-Tlak
- web aplikacija e-Tlak

SAŽETAK

Aplikacija e-Tlak namijenjena je praćenju krvnog tlaka i srčane frekvencije kao osnovnih podataka u dijagnostici arterijske hipertenzije. Čini ju mobilna aplikacija e-Tlak i web aplikacija e-Tlak. Rad se dijeli na teorijski i praktični dio razvoja aplikacije. Teorijski dio obuhvaća sve vezano za hipertenziju kao zdravstveni problem današnjice te detalje o korištenim tehnologijama vezanim za implementaciju dviju aplikacija. Praktični dio odnosi se na izradu mobilne i web aplikacije. Prvenstveno, mobilna aplikacija e-Tlak omogućuje unos krvnog tlaka putem mobilnog uređaja, a web aplikacija e-Tlak omogućuje vizualizaciju podataka unutar internetskog preglednika. E-Tlak pospješuje promicanje zdravlja i prevenciju kardiovaskularnih bolesti.

Ključne riječi: *krvni tlak, arterijska hipertenzija, mobilna aplikacija, web aplikacija, tehnologije*

SUMMARY

The e-Tlak application is intended for monitoring of blood pressure and heart frequency as basic data in the diagnosis of arterial hypertension. It is about e-Tlak mobile application and e-Tlak web application. The paper is based on the theoretical and practical part of the application development. The theoretical part covers everything related to hypertension as a health problem nowadays and details about technologies used for implementation of two applications. The practical part refers to the development of mobile and web applications. Firstly, the e-Tlak mobile application allows inputting the blood pressure data via a mobile device and the e-Tlak web application allows visualization of data within the Internet browser. The E-Tlak helps promote the health and prevention of cardiovascular diseases.

Keywords: *blood pressure, arterial hypertension, mobile application, web application, technologies*